Precision Navigation Using Pre-Georegistered Map Data

THESIS

Frederick Webber, B.S.M.E.C.S.

AFIT/GE/ENG/09-54

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/09-54

Precision Navigation Using Pre-Georegistered Map Data

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Frederick Webber, B.S.M.E.C.S.

September 10, 2009

AFIT/GE/ENG/09-54

Precision Navigation Using Pre-Georegistered Map Data

Frederick Webber

B.S.M.E.C.S.

Approved:

LtCol Michael J. Veth, PhD (Chairman)

3 SEP 09
date

Dr. John F. Raquet (Member)

2 SEP 09
date

Dr. Gilbert L. Peterson (Member)

2 SEP 09
date

AFIT/GE/ENG/09-54

Abstract

Navigation performance in small unmanned aerial vehicles (UAVs) is adversely affected by limitations in current sensor technology for small, lightweight sensors. Because most UAVs are equipped with cameras for mission-related purposes, it is advantageous to utilize the camera to improve the navigation solution. This research improves navigation by matching camera images to a priori georegistered image data and combining this update with existing image-aided navigation technology. The georegistration matching is done by projecting the images into the same plane, extracting features using the techniques Scale Invariant Feature Transform (SIFT) [5] and Speeded-Up Robust Features (SURF) [3]. The features are matched using the Random Scale and Consensus (RANSAC) [4] algorithm, which generates a model to transform feature locations from one image to another. In addition to matching the image taken by the UAV to the stored images, the effect of matching the images after transforming one to the perspective of the other is investigated. One of the chief advantages of this method is the ability to provide both an absolute position and attitude update.

Test results using 15 minutes of aerial video footage at altitudes ranging from $1000m$ to $1500m$ demonstrated that transforming the image data from one perspective to the other yields an improvement in performance. The best system configuration uses SIFT on an image that was transformed into the satellite perspective and matched to satellite map data. This process is able to achieve attitude errors on the order of milliradians, and position errors on the order of a few meters vertically. The along track, cross track, and heading errors are higher than expected. Further work is needed on reliability. Once this is accomplished, it should improve the navigation solution of an aircraft, or even provide navigation grade position and attitude estimates in a GPS denied environment.

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
UAV	Unmanned Aerial Vehicles	iii
SIFT	SIFT	iii
SURF	SURF	iii
RANSAC	RANSAC	iii
GPS	Global Positioning System	6
WGS-84	World Geodetic System of 1984	6
ECEF	Earth-centered Earth-fixed	8
DCM	direction cosine matrices	13
GPU	Graphical Processor Unit	85
pdf	Probability Density Function	91
UAV	Unmanned Aerial Vehicles	1
SIFT	SIFT	1
SURF	SURF	1
RANSAC	RANSAC	1

PRECISION NAVIGATION USING PRE-GEOREGISTERED MAP DATA

I. Problem Statement / Overview

Navigation is the act of estimating the position, velocity, and attitude of an entity, such as a vehicle. Humans and animals are naturally able to navigate, but getting a machine to do this poses many challenges. Much headway has been made with the advent of the Global Positioning System (GPS) and the development of inertial (motion) sensors. By tying these together, precise navigation is possible. Given enough power and payload capacity, these sensors can be designed to provide a highly accurate navigation solution. However, if power and weight or space capacity becomes scarce, as is the case on small unmanned vehicles, compromises are made to preserve power and payload cost. The inertial sensor is especially vulnerable to this, as limitations in current technology cause it to become substantially less accurate. Because of its removal, the performance of the navigation sensors suffer. These performance issues manifest as a greater uncertainty in the position, velocity, and attitude.

For some operations, this magnitude of uncertainty is not a problem. If the unmanned vehicle is tasked with basic intelligence gathering or with delivering externally targeted munitions, it can still relay pictures or accomplish its destroy mission quite well. However, in order to carry out more complex missions, such as taking pictures AND relaying coordinates, or spotting a target AND determining the target's location, it must have significantly more precise navigation. This is because the errors in estimated position and attitude of a target in an image are greatly exacerbated by the distance between the vehicle and the ground. Targeting in this way is known as geolocation. Even small errors cause a large problem in geolocation. For example, an error of 1^o in the roll or pitch at an altitude of one kilometer can produce a target geolocation error greater than 15 meters. If this information is being used to generate a tactical map or to select coordinates for bombing, it will have a tremendous impact on mission success.

For this reason it is very desirable to improve the performance of the navigation of these vehicles. The best solutions will require negligible additional power or payload burden. However, the most significant changes will be realized with the addition of another sensor. For this reason, the camera, which is onboard many of these aircraft, is exploited to serve double duty as both intelligence and as a sensor for navigation. The potential for improving the navigation solution, and thus the geolocation capability of the vehicle, is explored in this thesis by matching a priori map data to the current view of the camera.

The use of a priori map data has a significant valuable implication: it can be used in conjunction with the vehicle camera to provide an absolute position and attitude measurement. The inertial sensor can only provide differential measurements, so the uncertainty in the current position never decreases. Currently, the GPS is the only sensor that can provide an absolute position measurement. This is a primary reason why a denial of GPS attack is a tremendous risk. This work has the potential to provide a high-quality absolute position by matching the images from the vehicle camera to the registered map data. Such an update may serve as a substitute for GPS in its absence. This notion is formulated more precisely in the following section.

1.1 Problem Definition

Exploiting the camera using a priori data requires three components: a camera (image sensor), the a priori data itself, and the ability to match and process the data provided by the two.

1.1.1 Camera. The camera onboard the vehicle provides a representation of the physical appearance, position and orientation of the environment around the vehicle in the direction the camera is pointing. Distinct features in the terrain, such as roads or buildings, are present in the physical appearance information. This information is is used to match to the appearance of similar distinct features present in the maps collected to the features compiled into the a priori data.

1.1.2 A Priori Data. Using a camera, data can be collected of an intended theater of operation ahead of time. Because of the probable high threat in this area, this data will likely be provided by satellites and other aircraft. In the case of satellites, images can be taken of a given region visible from its orbit every time the satellite passes. This data can be used to build high precision maps. This high precision data is commercially available as well as from the National Geospatial-Intelligence Agency and the US Geological Survey organization [2] [1]. Using these maps, features contained within the images can be assigned a precise location.

Distinct features detected in the a priori data are compared against the information recorded by the cameras. The relative position and orientation of the features detected by the camera on the vehicle can be matched to those in the database, which have a precise position assigned to them. This information can then be used to correct the attitude and position of the vehicle.

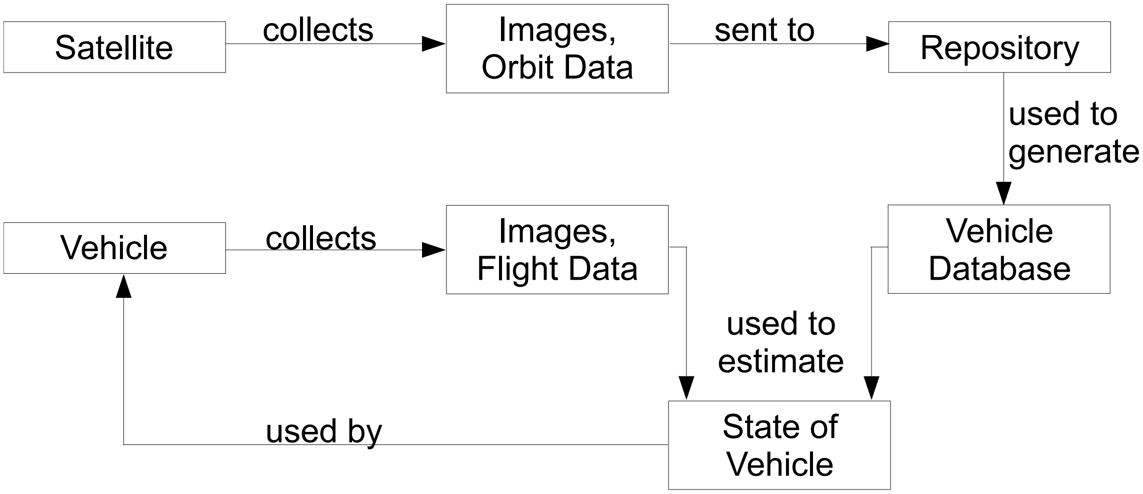


Figure 1.1: The image data from the satellite and vehicle are analyzed and combined in this research to develop an estimate of the state of the vehicle (that is, its position and attitude). The database itself is generated elsewhere by an intelligence organization or other company with access to such satellites. The region of interest is acquired and pre-processed for use in the onboard database.

The next section specifically describes the contributions of this thesis to the field of guidance, navigation, and control.

1.2 Contributions

As a result of this work, several contributions are made. The primary contribution is the development and analysis of utilizing surveyed a priori image data in conjunction with a camera onboard a small vehicle. This process developed is described in Chapter III and the results are presented in Chapter IV.

Directly supporting the primary contribution are two others, which are nearly as significant as the primary one. The first is the enhancement of the reliability of matching point features. It should be noted that it is virtually impossible to guarantee that a feature from one image matches the feature in the other image. This can be demonstrated by finding a very repetitive surface, such as a top-down view of a desert or grassy plain or even a plain wall and photographing it in two different locations. Matches will be found between the two images, and none of them will be correct. Conversely, other than the trivial case of photographing the same location twice consecutively without changing position or attitude and with negligible change in lighting, a perfect match is very highly improbable. By photographing the same object or location from even a slightly different position, attitude, or lighting condition and attempting to detect and match features, many correct matches will be made. However, it is virtually impossible for any of them to be perfect matches. It is desirable, then, to find a realistic way of increasing the reliability and integrity of the matches. This is accomplished using a weighting system, which is motivated in Chapter III and discussed with additional detail in Appendix A.

Lastly, the relative position and attitude estimator RANSAC's repetitive nature is exploited to create a statistically-based estimate of the vehicle state. This is described in Chapter III and rationalized in Chapter IV. Because RANSAC makes multiple attempts to estimate the best relative translation and rotation between two images, it takes little additional computation power to record each of these. By applying the weighting system involved in the reliability improvement, the weighted statistical mean and uncertainty are found. This provides a substantially more reliable solution than simply using the best solution.

1.3 Document Structure

The remainder of this thesis is laid out as follows: Chapter II presents the mathematical and theoretical groundwork for this research. It begins by definining the reference systems utilized in this document. Next, it briefly describes the sensors that provide the information needed for navigation. It then broadens in scope to cover other details, including image-based navigation techniques and briefly surveys other relevant works in the field. Chapter III develops the navigation and geolocation algorithms incorporated in this work. It also describes the simulations which produce the data presented in Chapter IV. In Chapter V, concluding remarks are made regarding the feasibility of deploying this work as well as future development that should be done. The appendices present additional relevant information.

II. Mathematics and the State of the Art of Geolocation

Geolocation is the science of estimating the position of an object on the Earth. This can be accomplished in a number of ways, including using images. This particular study is applied to images taken from an aerial perspective using a camera attached to an aircraft (henceforth 'agent'). The purpose of this chapter is to provide the necessary background to explain geolocation and the proposed improvement, a georegistration-based update to the navigation system. Georegistration is the act of identifying an object (typically in an image) and recording its surveyed position. Figure 2.1 shows an image taken from an aircraft along with a corresponding georegistered image, which is used for such a correction.

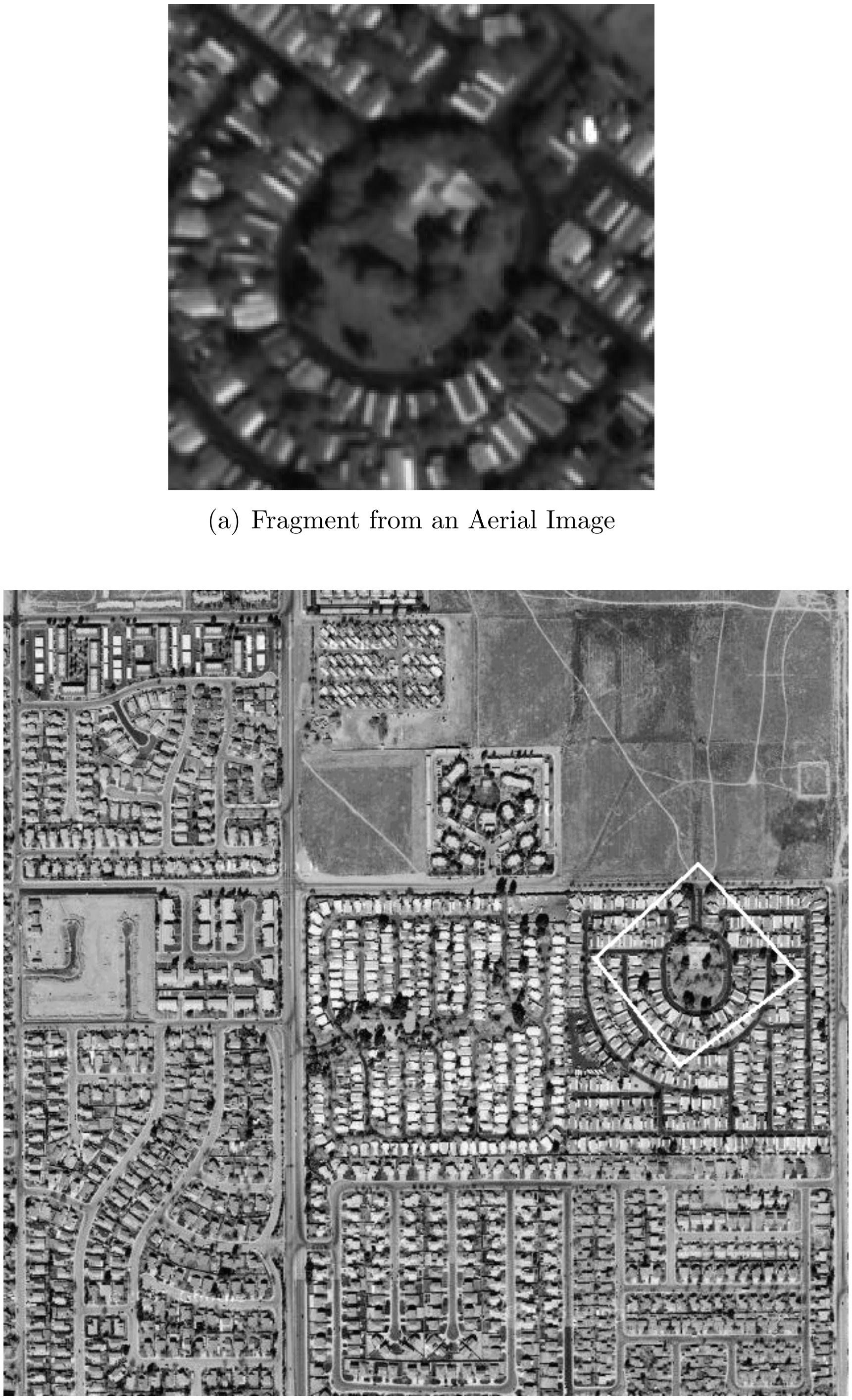
This chapter begins by covering the pertinent background needed to develop such an algorithm. The first section will cover coordinate systems, so that measurements may be compared sensibly. Following is a brief overview of navigation and an estimation tool, the Kalman filter, as well as the development of the Extended and Unscented Kalman Filters. Sections 2.3 and 2.4 cover the primary sensors augmenting the onboard Global Positioning System (GPS), where the primary objective is to explain the utility and drawbacks of inertial navigation and to present the basics of the use of cameras. With those in place, a complete explanation of the process of geolocation is possible, and is given in Section 2.7.

After the mathematical background, select developments in geolocation are explored, followed by an explanation of select computer vision techniques in feature detection and description. These features are used with georeferenced features to provide an update to the navigation solution.

2.1 Coordinate Systems

This paper utilizes several different reference systems. Reference systems describe the position and orientation of a body or a vector relative to a specified datum.

With the exception of the World Geodetic System of 1984 (WGS-84), each of these systems are cartesian, meaning they have an orthonormal basis in $\Re^3$. Inertial



(a) Fragment from an Aerial Image

(b) Georegistered Database Image

Figure 2.1: These two images demonstrate some of the basic differences between images taken by the aircraft 1(a) and images from the satellite 1(b). Satellite imagery is generally oriented north as the top of the image. It generally appears relatively flat. Each image is taken from the same altitude. Images taken from an aircraft are aligned with its attitude, causing the top of the image to usually not be north, and the roll and pitch is reflected in the apparent warping of the image.

frames are any non-accelerating, non-rotating reference where Newtonian mechanics apply. Inertial frames are assigned an origin and orientation suitable for its purpose. Navigational frames move with any rigid body they are attached to [7].

The last part of this section describes how to transform from one cartesian reference frame to another.

2.1.1 WGS-84. The World Geodetic System of 1984 describes positions relative to the center of the Earth using latitude, longitude, and altitude. This is shown in Figure 2.2. Altitude is relative to a defined ellipsoid, and conveyed in meters. Latitude and longitude are similar to spherical coordinates. Latitude is zero degrees at the equator. It increases or decreases in value to a maximum of 90^o in parallel circular cross sections north and south. Longitude is set to 0^o at the IERS Reference Meridian, which runs near Greenwich, England. It is counted to 180^o west and east from that meridian [10].

Global geodetic systems must define two surfaces. These surfaces are a mathematical surface of reference called the ellipsoid and an equipotential surface called the geoid [10]. As the ellipsoid is the mathematical basis of the system, it is important to define it. The semi-major axis of the ellipsoid has a radius of 6378137.0 meters, and a flattening factor of $\frac{1}{f}$=298.257223563 [10].

2.1.2 Earth-Centered, Earth-Fixed (ECEF) Frame. The Earth-centered Earth-fixed (ECEF) frame has an origin located at the center of mass of the earth. The x axis originates at the center of the Earth and points out through the equator where it intersects the Prime Meridian. The z axis also originates at the center of the Earth and points through the North Pole. The y axis completes the right handed set of axes [10]. This frame is subsequently refered to as the e frame.

2.1.3 Body Frame. The agent's kinematic equations are generally computed using the body frame, which is centered on the vehicle's center of gravity and denoted

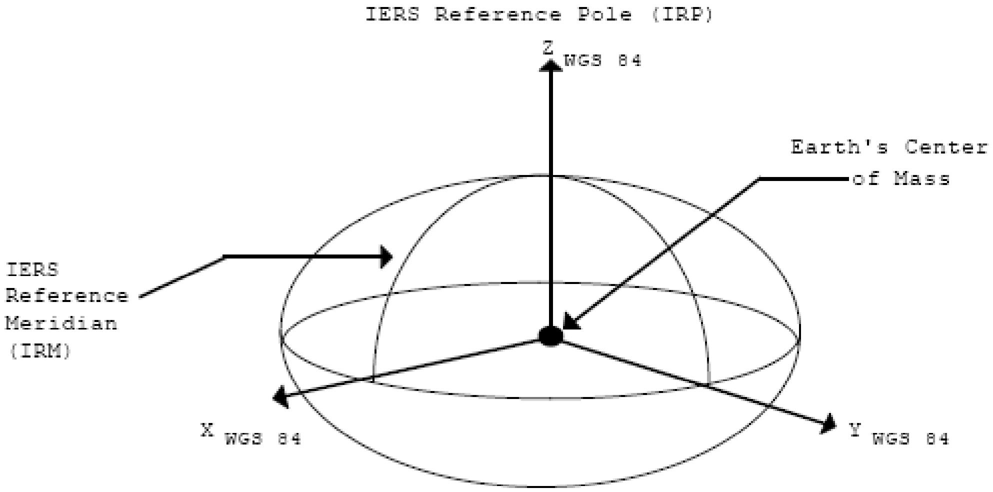


Figure 2.2: This shows the WGS-84 Reference System [10]. The ellipsoid shown is not necessarily coincident with the surface of the Earth.

with the sub and superscript b. The body frame (b frame) points the x axis out the nose or front of the agent, the y axis out the right side, and the z axis down.

2.1.4 Navigation Frame. The navigation frame is fixed to the earth. The orientation and initial position must be pre-specified. Generally, it is initialized to be identical to the body frame when the agent is in a stable, non-moving state, before deployment. The n frame is generally referenced to the b frame in terms of the angular offset about the x-axis (roll), y-axis (pitch), and the z-axis (yaw). These angles are known as Euler angles. This frame and the e frame are shown in Figure 2.3.

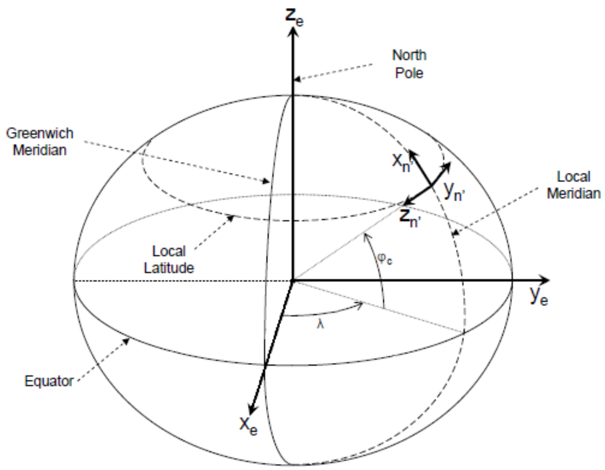


Figure 2.3: The Earth-centered Earth-fixed and navigation frames are both fixed to the earth. While the ECEF frame is always the same, the navigation frame can be adapted and re-initialized as needed. Adapted from [7].

2.1.5 Camera Frame. The camera frame is expressed with z being normal to the center of the image field, while the x and y axes point out the top and right of the image plane, respectively. For this research, the camera points in the nadir direction of the aircraft, that is, the z axis points the same direction of the z axis in the navigation frame and the y axis points in the opposite direction of the y axis in the navigation frame of the aircraft. The camera frame is illustrated in reference to the camera in Figure 2.4 and in reference to the image plane in Figure 2.5.

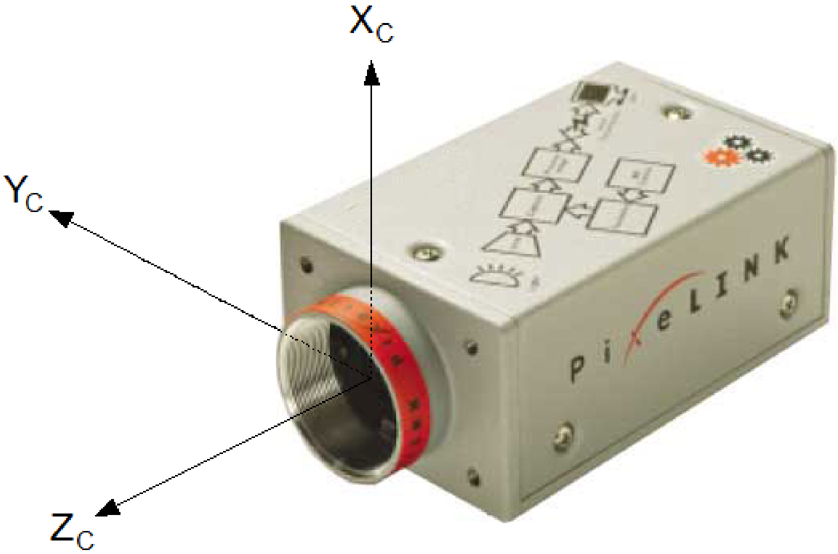


Figure 2.4: The camera frame is used to relate images taken by the camera to other reference frames.

2.1.6 Image Plane. The image plane is not really a reference frame in quite the same sense as the others, because it is essentially a two dimensional surface with definite bounds. Pixels are discrete values as well, though they can be interpolated to obtain fractional values. The image plane relates to the camera frame as follows: the camera frame has an origin at the center of the image, and extends to the third dimension by having positive z originate at the center and progress away from the camera body, coaxially with the lens (i.e., perpendicularly to the image). A diagram relating the two is shown in Figure 2.5.

2.1.7 Coordinate Transformations. Reconciling and comparing locations requires that they be specified in the same system - i.e., have the same units and the

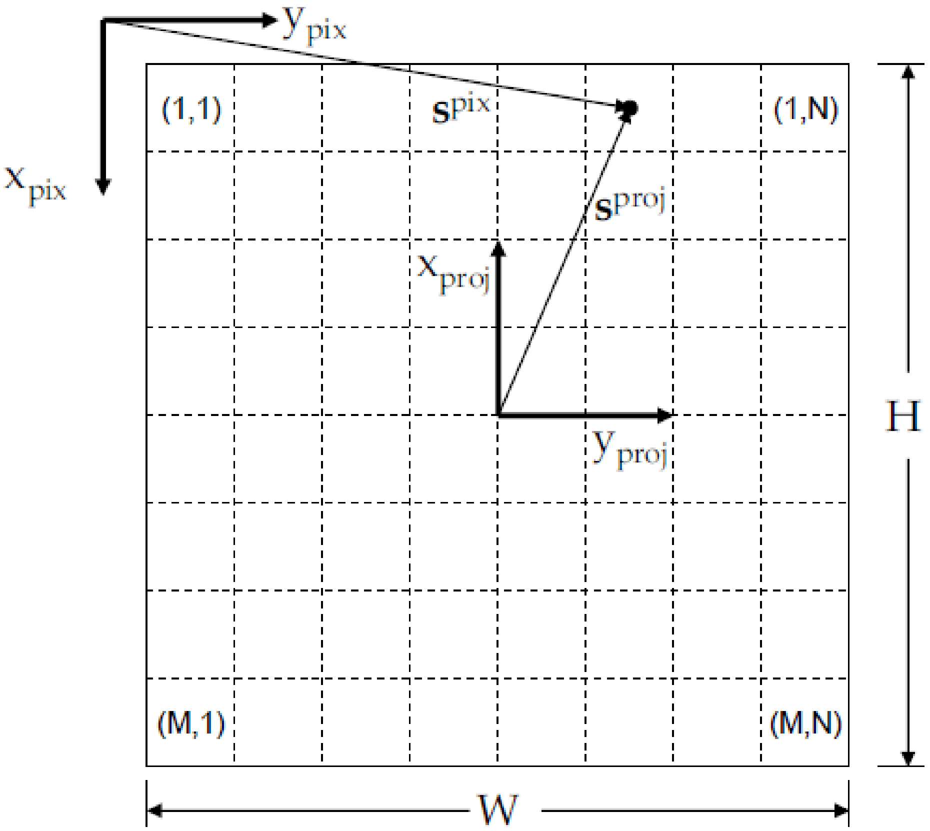


Figure 2.5: This shows the relationship between the camera frame and the image plane. The *proj* subscript is the conversion from matrix notation (in pixels, *pix*) to the camera frame. The *proj* component, also measured in pixels, is rescaled to achieve the *c* frame. The scaling depends on the altitude and the camera and lens calibration. [7].

same axes. This is simple for each of the cartesian systems. The WGS-84 specification includes directions for converting to the ECEF frame, so it is equivalent to a cartesian system.

It is useful at this point to describe a transformation to enable conversion between each of these frames. The conversions are done by rotating and translating from one frame into another. This is done with direction cosine matrices (DCM). By applying a DCM to a vector, the vector is adjusted to reflect a different orientation, but its magnitude is unchanged. Equation (2.1) shows vector j transformed by a DCM from abstract frame j to another abstract frame k. The DCM itself is represented by C, in this case, C_j^k. The vectors provide the same information, but for different reference systems. The position is represented as s, which is shown as s^k for the position in the k frame and s^j for the j frame.

$$s^k = C_j^k s^j \tag{2.1}$$

With a set of common reference systems in place, it is now possible to extract sensible data from the various incorporated sensors and make use of them.

2.2 GPS

Latitude, longitude, and altitude can be determined by a global positioning system (GPS) receiver. Position is determined by computing the range to four or more GPS satellites. The range to a satellite is determined by subtracting the time the signal was sent by the satellite, which is indicated in the transmission, from the time it was received. Like any other measuring system, errors must be addressed. The principal error is the receiver time bias, which requires the fourth satellite. Other primary errors include delays introduced by the troposphere, ionosphere, and from multipath (signals reflecting off nearby surfaces). A horizontal error of $1.8m$ (1σ) can be achieved [15].

2.3 Inertial Measurment Systems

Inertial sensors are able to detect and measure motion in six degrees of freedom.

2.3.1 Background and Function. The proliferation of inertial sensors has been made possible largely due to the development of micro electrical mechanical systems (MEMS). Inertial sensors have seen use in guidance, navigation and control systems for most types of mobile platforms [13].

A six degree inertial sensor features three orthogonal accelerometers along with three orthogonal gyroscopes. The measurements from the gyros provide angular rate while the accelerometers provide specific force. These measurements are then integrated to provide information about position, velocity and attitude and to resolve these components into the navigation frame. Gyroscopes provide information on the rate of vehicle turn with respect to the inertial frame, but accelerometers are unable to distinguish total acceleration of the vehicle with respect to inertial space from gravity. Calculations must be made using knowledge of gravity in order to correctly resolve the vertical acceleration [13].

In an ideal world, the inertial sensor would give perfect information and there would never be any uncertainty or error in the position, velocity, or attitude estimates or the associated measurements. However, in practice, inertial measurement units are subject to biases and errors. Some of these fluctuate in an unpredictable manner such that they cannot be compensated for, whereas others can be eliminated and cancelled out via calibration and tuning techniques.

Manufacturing imperfections create the host of major sources of bias and error listed at the end of this section. Each error will generally consist of fixed or otherwise repeatable terms as well as unpredictable variations in the sensor. These can be due to temperature and variations between applications of power, among other causes [13].

Inertial measurement units or their associated filters (or both) must incorporate a negative feedback loop to handle instability in the vertical orientation, or else even the most minute offset in expected and actual gravity will quickly telescope the estimated velocities to infinity. In addition to this, there are several biases and errors inate to accelerometers. Errors in velocity and position propogate over time due to angular alignment errors and due to the accumulation of small errors summed together [13]. Figure 2.6 demonstrates the relationship between the data gathered by the sensors and the computable outputs.

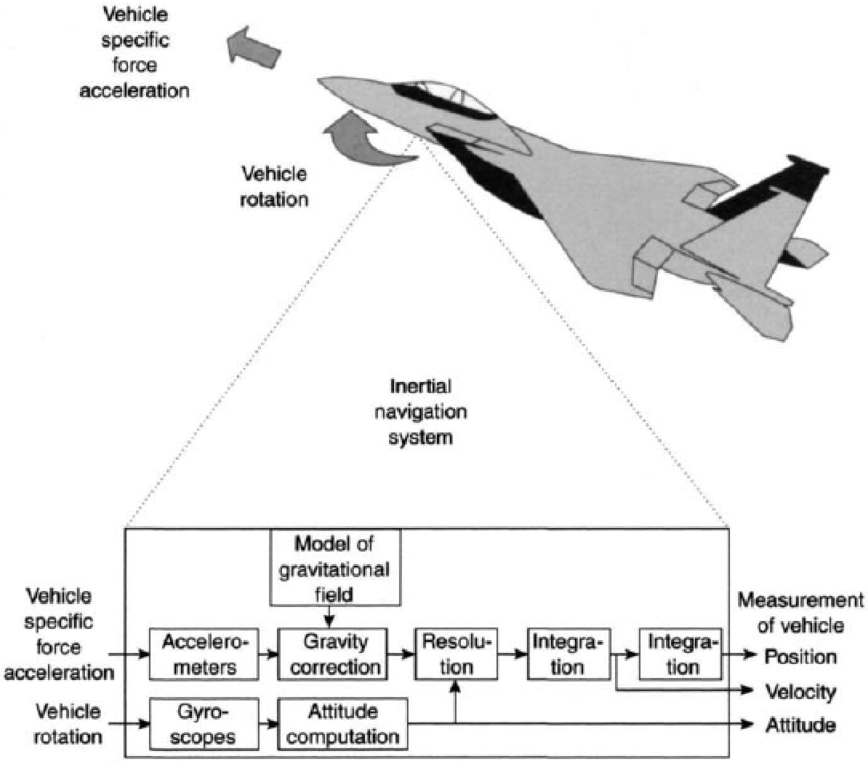


Figure 2.6: The inertial sensor measures vehicle specific force acceleration and rotation. [13]

The rest of this section explains some of the more prominent biases and errors, which is a significant motivating factor behind the need to improve the navigation solution. It is split into three parts. The first part covers the biases and errors that both the gyroscope and the accelerometers experience, and the other two are for biases and errors unique to that portion of the sensor. These errors are much more exaggerated in smaller agents, such as the UAVs that are the target of this research.

2.3.2 System-wide Biases and Errors. This portion covers biases inherent in both accelerometers and gyroscopes.

- *Fixed bias*: Sensors, even when static relative to the Earth, will often indicate motion when indeed there is none. A variety of effects may induce a fixed bias, ranging from temperature or magnetic field gradients to manufacturing defects and imperfections. A fixed bias manifests itself when the system is static, and yet the sensor reports motion. The magnitude and direction of the bias is independent of any actual motion. For the accelerometers, this bias is generally reported in milli-g or micro-g, depending on precision. For gyroscopes, this is generally reported in of degrees per hour $\left[\frac{\text{deg}}{h}\right]$.
- *Scale-factor errors*: Nonlinearities may exist within the sensor, such as nonlinear resistances, that cause the output to not use a constant ratio of the measured input to the described output. These nonlinearities are in part due to manufacturing flaws or leaving the domain in the specifications for which the output is considered valid. This type of error refers to systematic deviations from a least-squares line fit to the measurements and is described in parts per million [ppm].
- *Cross-coupling errors*: Cross-coupling occurs when sensors are not truly orthogonal to each other. This is due to manufacturing error, and it too can be tested and determined. This can be detected in testing and is reported in parts per million [ppm] or as a percentage.

2.3.3 Gyroscopic Biases and Errors. This portion describes biases and errors that are either unique to the gyroscope or, if present at all in the accelerometer, are insignificant.

- *Acceleration-dependent bias*: Applied accelerations, often due to gravity, are capable of introducing a proportional amount of bias into the system. This occurs when the center of gravity and the center of suspension are not coincident.

The primary ways of this error occurring are when such accelerations occur both along and orthogonal to the axis of rotation. Acceleration-dependent bias is generally reported in degrees per hour, per g $\left[\frac{^o/h}{g}\right]$.

2.3.3.1 Accelerometer Biases and Errors. The error listed here is unique to the accelerometer, though it is comparable to the acceleration-dependent bias listed for the gyroscopes.

- *Vibro-pendulous errors*: Cross-coupling can occur in aligned sensors when the pendulum internal to the sensor experiences an angular displacement. The error is most severe when the vibration occurs in a plane that is normal to the pivot axis and at 45^o to the sensitive axis. This error is expressed in units of $\left[\frac{g}{g^2}\right]$.

The GPS and inertial sensor are not the only sensors that can be used to measure position and motion. Key to this project is the utilization of the imaging sensor, or camera, which is presented next.

2.4 Camera

The camera, or image sensor, is a device typically found onboard current unmanned aerial vehicles being used for reconnaissance. Since the device is already onboard, this research will not require any additional weight, wiring, power, etc., it will simply utilize what is already there.

In order to compare a pixel from an image captured by a camera to an object in the 'real world', any distortion applied by the lens must be removed, and then the image needs to be rescaled. The results and explanations in this section are adapted from [7], where they are treated much more in-depth.

2.4.1 Lens Distortion. Lenses can introduce several different kinds of distortion. However, this research only attempts to treat the severe tangential distortion from the fisheye lenses. This causes what should be straight lines in the image to appear as rounded. The error is modeled as increasing as the distance from the center of the image increases, or a radial distortion.

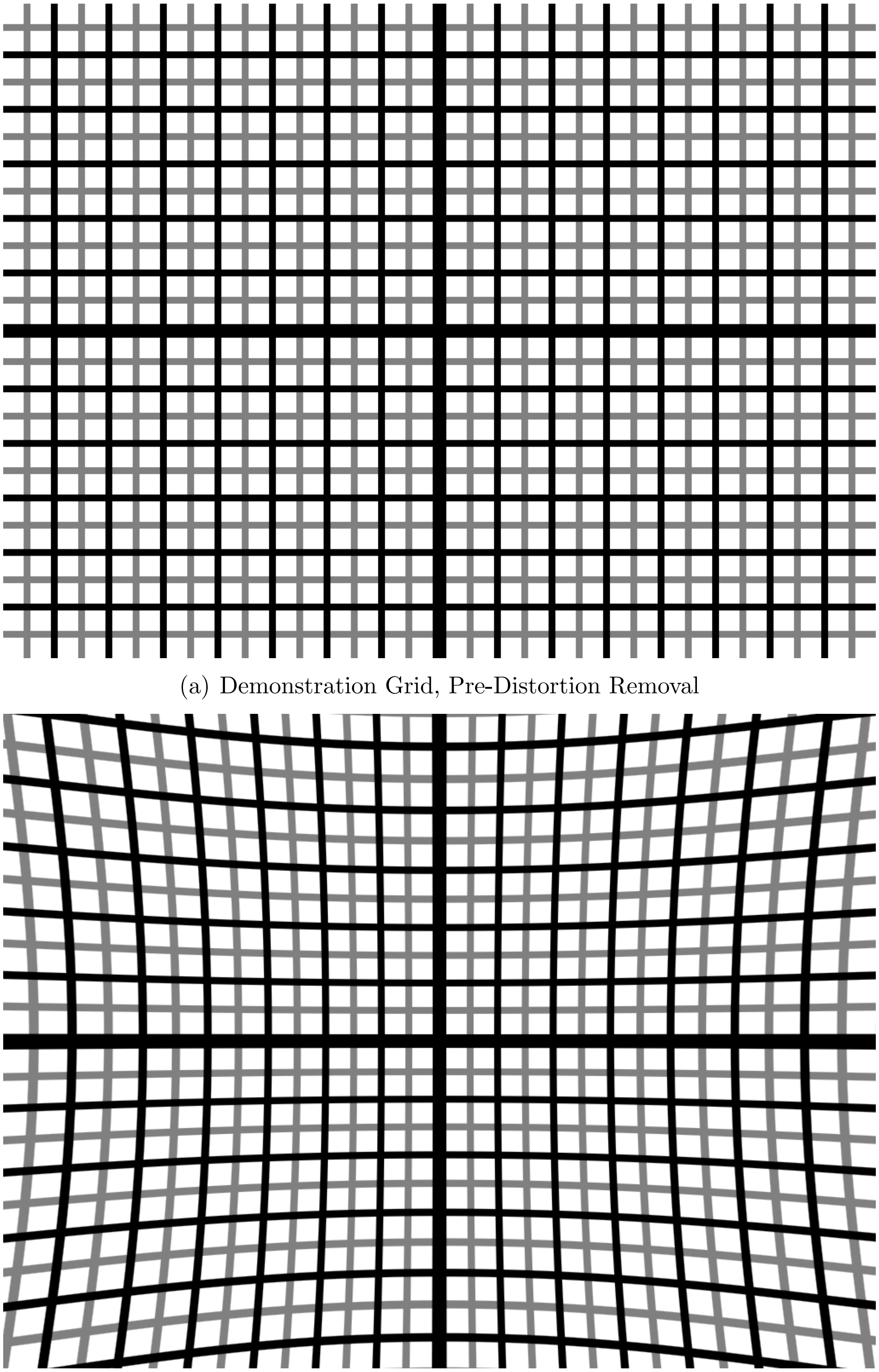
The distortion removal algorithm implemented takes the pixels in the original image and re-assigns the location of these pixels. This leads to pixels not being located at integer locations, so the algorithm interpolates to find new values for the remaining pixels. For this project, the outside part is removed. See Figure 2.7.

Lens distortion can be estimated by a model or determined more precisely on a pixel-by-pixel basis. This project utilized a generalized model for the whole camera-lens combination rather than a pixel-by-pixel transform. Such a model can be developed via calibration techniques that are beyond the scope of this document. Calibration is used to develop a model of both lens distortion and scaling information. See [7] for further details.

Figure 2.7 shows what happens if the same undistortion procedure that is applied to the real images is applied to a grid. The goal was to show that part of the original image was lost and that the image undergoes a sort of 'shrinking' inward. Figure 2.8 shows a raw camera image of Lancaster, California. Note that the roads, which run east and west or north and south (the image was taken with a northeast bearing) are all arced in an abnormal manner. After distortion is removed, the roads in the image are now straight. Some parts of the image have been lost off to the side, as it was trimmed to not show any void space and to maintain the original size as much as possible.

This work is slightly inconsistent with the previous work done by [7] because image distortion is removed prior to feature detection (explained in Section 2.8).

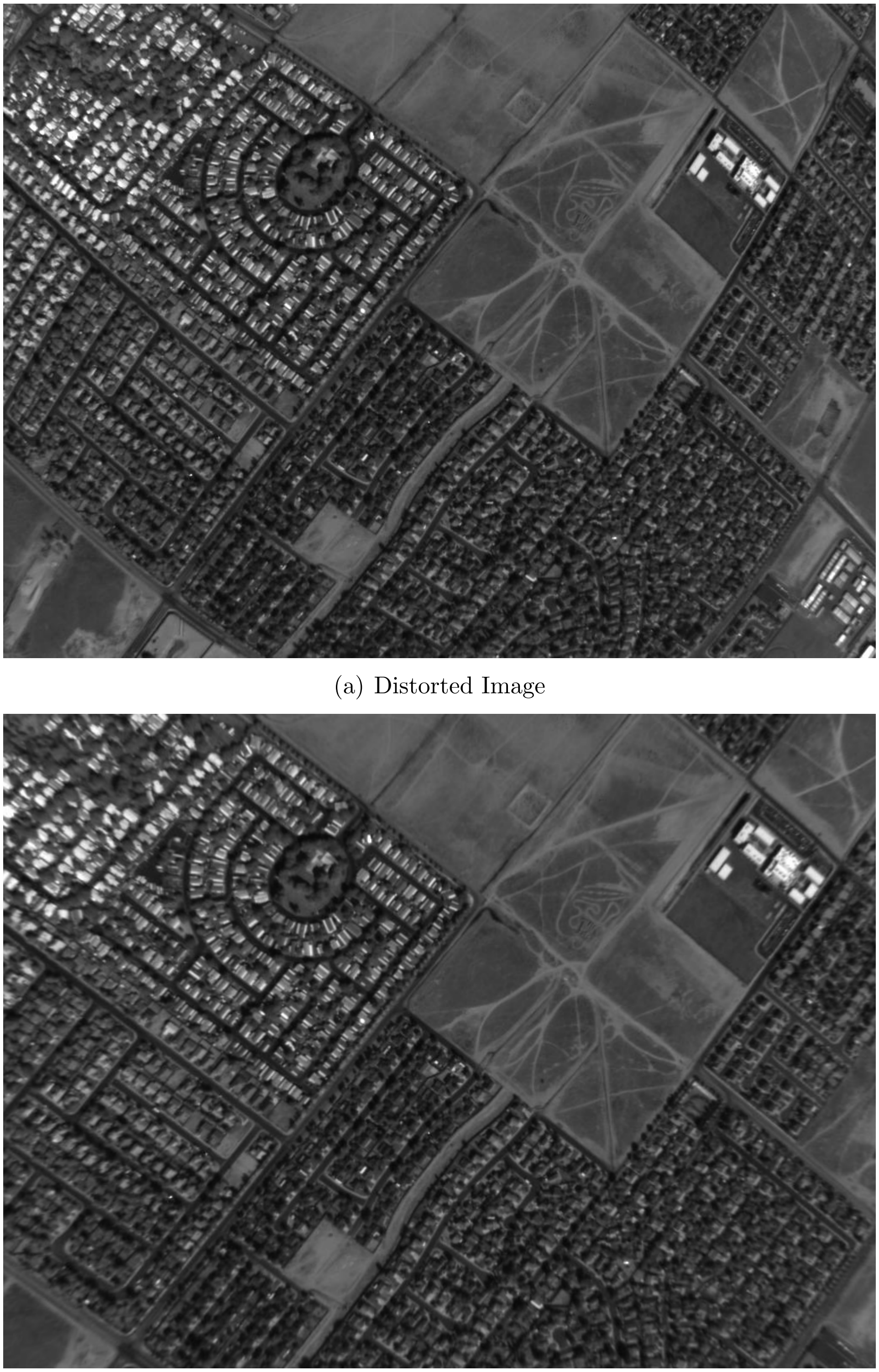
2.4.1.1 Camera Parameters and Scaling. No image in this research is life-sized, but is instead scaled. Accordingly, each pixel represents so many meters. A transform needs to be developed, then, to convert pixels from the image plane to what is called the Camera Frame, or c frame. The parameters listed in Table 2.1 are of interest in developing such a transform.



(a) Demonstration Grid, Pre-Distortion Removal

(b) Demonstration Grid, with Distortion Removed

Figure 2.7: The de-warping process causes parts of the image to stretch and deform, relative to the original. The goal, however, is to attempt to recreate a more realistic image.



(a) Distorted Image

(b) Undistorted Image

Figure 2.8: Lens distortion causes an image to appear warped.

Table 2.1: List of Parameters For Camera Coordinate Transformation

Parameter	Symbol	Units
horizontal resolution	N	pixels
vertical resolution	M	pixels
image plane width	W	meters
image plane height	H	meters
focal length	f	meters
offset from center of gravity	p_b^{cam}	meters
orientation relative to agent	C_c^b	(DCM)

Table 2.2: Image Coordinate Notation

Parameter	Symbol	Units
pixel location	$\underline{s}^{pix}$	pixels
distance to the center of the scene	s_z^c	meters
c frame position	s^c	meters

These are combined into a single transform that, when applied to a vector of pixels (the depth of any pixel is 1), produces the normalized location in the camera frame or, when inverted, the location of the points in the camera frame when given the pixel location in the image. The transform of interest is

$$T_{pix}^{c} = \begin{bmatrix} \frac{-1}{f}\frac{H}{M} & 0 & \frac{-M-1}{2}\frac{-H}{M}\frac{1}{f} \\ 0 & \frac{1}{f}\frac{W}{N} & \frac{-N-1}{2}\frac{W}{N}\frac{1}{f} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

This can be used to develop several related vectors that describe the location of a point in any of the scene, image plane or the camera frame. Table 2.2 lists notation used to convert from image plane points to camera frame coordinates.

The point of all this nomenclature is the ability to take an undistorted pixel in the image plane and convert it to a point in a frame of interest, in this case, the e frame. When a picture is taken, the pixels are referenced in the image plane from the top-left pixel. To convert from the position of the pixel in the image plane $\underline{s}^{pix}$ to the normalized pixel location in the camera frame:

$$s^c = T^c_{pix} s^c_z \underline{s}^{pix} \tag{2.3}$$

Once a feature's location has been computed in the camera frame, it can then be transformed to the n frame or e frame, as needed. Equation (2.4) is used to transform the vector from the center of the camera to the position of the feature in the camera frame to the relative position of the feature to the navigation frame.

$$s^n = C^n_b C^b_c s^c \tag{2.4}$$

It is now prudent to develop a method of combining each of these sensors in as close to an optimal manner as possible. The next section presents such a method, the Kalman filter.

2.5 The Kalman Filter

The Kalman filter is an optimal estimator. A detailed derivation and proof of optimality, as well as the acting assumptions, are contained in [6]. The Kalman filter can be implemented in a continuous or discrete manner; here only the discrete version will be used. The Kalman filter maintains an estimate, $\hat{x}(t_i)$, of the state of a particular system, such as its position, velocity, and orientation, and a statistical description of these estimates in the form of the covariance $P(t_i)$. The state estimate represents the mean. The estimator works by taking measurements $z(t_i)$ and using the measurements to adjust the estimate and covariance in proportion to the optimal Kalman filter gain $K(t_i)$. The measurements are related to the states by the association matrix $H(t_i)$.

To propagate the values to the next time step, the matrix $B(t_i)$ associates the inputs $u(t_i)$ with each variable in the truth state vector $x(t_i)$. Likewise the matrix $G(t_i)$ relates the strength of the covariance of the inputs $Q(t_i)$.

At each time step in the discrete computation, the previous state is propagated to the next time increment. Then, these vectors and matrices are updated based on the current estimate (note that each matrix and vector is a function of time). Each update also considers the strength of the statistics expected noise $R(t_i)$.

The next part of this section describes the mathematics behind propagating and updating a linear Kalman filter.

2.5.1 Propagating and Updating the Kalman Filter. The Kalman filter has two tasks to perform to maintain a proper estimate of the navigation solution. One task is to propagate the state from one time to the next. For example, this includes updating the position based on the previous velocity estimate and the time since the last estimate. The other task is to perform a measurement update which corrects the state estimate based on feedback from the sensors. Between updates, it propagates the state based on the input from the inertial measurement unit. This can be corrected based on information provided by the GPS and any other sensors.

In the following equations, the superscript '-' indicates that the value has been propagated, but not updated. The times before and after propagation are designated as t_{i-1} and t_i, respectively. To propagate the state estimate and covariance from one time step to another:

$$\hat{x}(t_i^-) = \Phi(t_i, t_{i-1})\hat{x}(t_{i-1}^+) + B(t_{i-1})u(t_{i-1}) \tag{2.5}$$

$$P(t_i^-) = \Phi(t_i, t_{i-1})P(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + G(t_{i-1})Q(t_{i-1})G^T(t_{i-1}) \tag{2.6}$$

The matrix $\Phi(t_i, t_{i-1})$ represents the discrete time state transition matrix. The relation between the gain $K(t_i)$, estimate $\hat{x}(t_i^+)$, and covariance $P(t_i^+)$ is given by the following relations, where the superscript $+$ indicates the value after the update and $-$ indicates the value just before the update, both at the same time step t_i:

$$K(t_i) = P(t_i^-)H^T(t_i)\left[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)\right]^{-1} \tag{2.7}$$

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i)\left[z(t_i) - H(t_i)\hat{x}(t_i^-)\right] \tag{2.8}$$

$$P(t_i^+) = P(t_i^-) - K(t_i)H(t_i)P(t_i^-) \tag{2.9}$$

The initial estimate $\hat{x}(t_0)$ is generally set to known values, often zero if possible, and the covariance is set to the untertainty of the initial estimate, based on the measurement devices used to calibrate the system.

One severe limitation of the Kalman filter is that it is a linear estimator; that is, any system that must be modeled in a non-linear fashion will be estimated very poorly by a standard linear Kalman filter. To solve this problem, non-linear Kalman filters have been developed. They cannot mathematically claim optimality, but, depending on the quality of the model, they are near-optimal. Explanations are now given of the Extended Kalman Filter and the Unscented Kalman Filter, which are both non-linear estimators.

Next, the a non-linear estimator is presented.

2.6 Extended Kalman Filter

The basic mechanics of an Extended Kalman Filter are now presented. Like a linear Kalman filter, Extended Kalman Filters (EKF) assume all noise is additive Gaussian. Continual linearization is what separates the EKF from the linear Kalman filter. Linearization works as follows:

- Linearize the relational matrices Φ, H, B, G, and Q by computing the Jacobian, that is, the partial of each listed matrix with respect to each variable present.

- Prior to propagating or updating, compute the value of the above matrices at time t_i (or time (t_i, t_{i-1}) as appropriate) by substituting the estimated value of the state vector into the Jacobian.
- The Extended Kalman Filter focuses on differences between estimated and measured values. Updates are computed in terms of the residual between the estimated state and the measured state.

The strengths of the noises are denoted as Q for $\mathrm{w}(t_i)$ and R for $\mathrm{v}(t_i)$. The matrix G is the mapping between the white noises and which states are actually corrupted by them.

Extended Kalman Filters are susceptible to linearization errors. All Kalman filters are susceptible to modeling errors, but the EKF exaggerates these issues due to its linearization. Because of the linearization, it is at higher risk of a diverging solution.

2.6.1 Propagation. Extended Kalman Filters operate by estimating the difference between the true value, which are estimated by the filter, and the nominal value, as determined by measurements. The rate at which this changes is

$$\delta\hat{x}(t) = \hat{x}(t_i) - \bar{x}_n(t_i) \tag{2.10}$$

In an Extended Kalman Filter, $\bar{x}_n(t_i)$ represents the estimate of the state based on the linearized state transition matrix and the previous state. Equation (2.10) becomes

$$\delta\dot{x}(t) = F\left[t, x_n(t)\right]\delta x(t) + G(t)w(t) \tag{2.11}$$

In the above equations, F is the linearized state transition matrix.

2.6.2 Performing an Update. The measurement is similarly predicted, and the difference between the propagated value (believed to be true), and the actual measurements is used to perform all needed updates. Because of the non-linear nature of the model, the measurements are also linearized. An update is performed as follows:

$$K(t_i) = P(t_i^-)H^T\left[t_i, \hat{x}(t_i^-)\right]\left[H\left[t_i, \hat{x}(t_i^-)\right]P(t_i^-)H^T\left[t_i, \hat{x}(t_i^-)\right] + R(t_i)\right]^{-1} \tag{2.12}$$

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i)\left[z_i - h\left[\hat{x}(t_i^-), t_i\right]\right] \tag{2.13}$$

$$P(t_i^+) = P(t_i^-) - K(t_i)H\left[t_i, \hat{x}(t_i^-)\right]P(t_i^-) \tag{2.14}$$

In the above equations, the matrix $\mathbf{H} = \frac{\partial h}{\partial x}|_{x=x_n}$. The update measurement at time t_i is given by z_i. Between updates, the state estimate is propagated through a numerical solver.

2.7 Geolocation

Geolocation is the science of estimating the position for an object on the earth represented in an image without being able to survey that point. In this thesis, geolocation is performed based on a point in an image taken by a flying agent and assigned a position. This section will focus on the mathematics and statistics involved in performing geolocation, as explained in [7]. Geolocation in this case is a function of the variables listed in Table 2.7.

Using the vector from the camera frame to the feature in the navigation frame, s^n, the location of the feature in the navigation frame of the feature (or target, t^n) can be established by:

Variable	**Symbol**	**Units**
Camera Position	p^n	meters
Camera Attitude	$\mathbf{C}_b^c$	DCM for orientation of camera relative to the body
Camera Distortion Model	(see 2.4)	*variable*
focal length	f	$\frac{meters}{pixels}$
distance to ground	s_z^c	meters

Table 2.3: This table lists various parameters of interest that are required for proper geolocation. As it stands now, it gives incomplete information and will be refined in subsequent sections, including Section 2.1 discussing coordinate frames as well as in Section 2.7, which explains in detail how Geolocation is performed.

$$t^n = p^n + s^n \tag{2.15}$$

Prior to computing this, the estimate of s^n will likely need to be refined using a local digital terrain elevation database (DTED) to estimate the height above the ellipsoid. The DTED is a mapping, generally by latitude and longitude, of the surveyed altitude of the ground at that point.

As simple as that may seem, the issue comes not in developing an algorithm to compute the location of the target, but in having a precision sufficient that the estimate is reliable enough for the task at hand, whether trying to develop a three-dimensional map or achieve a target-grade precision position solution. Using Equation (2.15) as a reference, the errors in t^n are directly related to any errors in s^n and p^n.

The errors in p^n are relatively small in an environment with sufficient GPS reception, especially when using a military receiver.

That leaves, then, the errors in s^n to be concerned with. The errors in s^n come from the attitude, position of the agent, and any errors in the DTED. The quality of the DTED can be improved by getting a newer version, a DTED with higher precision due to a finer granularity (i.e., more datapoints over a given area), and/or a DTED with greater accuracy. The quality of the position p^n could certainly be improved, but as will be demonstrated momentarily, it is far from the dominating term. As presented in [14], the horizontal target covariance is:

$$\sigma^2_{y_h} = \sigma^2_{p_h} + \left(\frac{1}{tan^2\bar{\theta}}\right)\sigma^2_{h_v} + \left(\frac{\bar{h}^2_v}{sin^4\bar{\theta}}\right)\sigma^2_{\theta} \tag{2.16}$$

In Equation 2.16, the overbar notation denotes the linearization value. Of these terms, it was established in [14] that attitude is the predominant error. The quartic term in the denominator of the third term, which is a function of the attitude estimate $\bar{\theta}$, tends to be dominant. Height is represented by h_v. This term is squared, and then multiplied by the covariance in the off-nadir angle.

The error terms related to the position and attitude of the agent are being simultaneously targeted by this work, so the results should show an improvement in both. The next section of this chapter discusses image processing techniques necessary to incorporate an update from the image sensor.

2.8 Feature Detection

In order to perform image registration, a robust, repeatable method of identifying features must be available. This procedure would be applied both to pre-recorded satellite or other intel images as well as the images captured during operation. While performing this sort of matching is nearly trivial for a human, it is a challenge in computer vision.

The simplest method to positively identify a point of interest is for the agent to capture an image that is pixel-for-pixel identical to one stored in a database. This is very unrealistic - the most immediately obvious reasons are that it is improbable to the point of being essentially impossible to take a picture from the exact same position with the exact same angle.

Beyond getting the same position and angle, many other factors come into play with trying to match images. Matching in that fashion requires near-perfect knowledge of the intrinsic parameters of any cameras involved.

Most of these issues, with care, are not a significant issue after calibrations and adjustments. The primary challenges remaining that can be realistically controlled through computational power are the orientation and scale components.

But, because such a scenario is so idealized and impractical, a more flexible approach is desirable. Many ways of handling this have been developed, including plane, edge, and point-feature detectors. The nature of this research lends itself very well to point features and very poorly to plane or edge-based features. Point features are generally found at distinctive locations, such as corners or T-intersections [3].

The first part of this section will be dedicated to overviewing the challenges presented in performing feature detection in two different images. The other two sections will discuss two algorithms for identifying a landmark and characterizing it so it can be located in a different picture of the same scene.

2.8.1 Challenges in Landmark Recognition. Landmark recognition is vulnerable to many types of problems. The areas of concern are split into two categories. The first category deals with how the cameras affect the images. The second deals with environmental factors that are largely uncontrollable.

2.8.1.1 Effects of Cameras on Landmark Recognition. The primary concern regarding the camera is developing a proper model. Knowledge of a camera model is required to compute a physical location for the object depicted in the image. The camera model provides information on the focal length of the camera as well as how many meters are represented by one pixel at that focal length. Unfortunately, the model for a camera is not permanent. If the focal length changes, if the lens on the camera is exchanged, or if any part of the image is out of the focal plane, the model degrades or becomes invalid. If the configuration of which specific camera, which specific lens (as each camera and lens is slightly different due to manufacturing flaws), and approximately what focal length the camera and lens were set to, these values can be computed using various calibration techniques. Having an accurate camera model was significant in this research. Camera models are pre-processing issues, that can be updated during operation if necessary.

2.8.1.2 Environmental Issues in Landmark Recognition. The environment of regions that have been photographed are not immutable; outdoor environments change almost constantly.

- **Illumination:** The level of illumination, so long as it is relatively uniform, is theoretically not a problem in this research, since the two methods used to detect features claim to be illumination-invariant over an unspecified domain. The domains can be thought of as having an amount of light comparable to any time between sunrise and sunset on a clear day. What is a tremendous problem with illumination, and this is in part coupled with orientation, is glare and substantial variations in reflectivity based on the angle. Illumination can be handled in part by choosing the time of day, weather conditions and seasons carefully. This, however, is a tremendous nuissance for reconnaissaince.
- **Orientation:** Orientation, discarding rotation parallel to the image plane, causes warping in the image, similar to what a rectangular map of the Earth does to Canada, Greenland, and Antarctica. It causes features visible from one angle to become obscured and can change the angle of incidence of light relative to the image sensor and thusly cause glare. Orientation, as noted in [5], includes an affine component, but affine-invariant transforms are generally inferior to scale and orientation-invariant transform unless the affine offset was around at least 40%. The next section describes two detectors used in this research that can be used to identify and match features.
- **Scale and Zoom:** Over a reasonable domain, both feature trackers are scale-invariant. However, at some point, it is possible to have too much zoom and defeat scale invariance. Consider what anything on the surface of the earth looks like in images taken from space; it would be impossible to identify anything smaller than a country without using a device to zoom in.

- **Time Invariance:** as time progresses, new buildings are constructed, old are torn down; natural disasters reshape land scapes; the changing of the seasons can completely change the appearance of the landscape, especially in areas where trees lose their leaves; even the time of day and weather has a very significant impact when considering reflectivity, interference, illumination sources and intensity, among other conditions. While identifying any given location even years apart is generally simple for a human, it is a tremendous challenge for a machine. Time Invariance can be handled by updating the source database regularly. This is handled by various intelligence agencies as well as commercial companies such as Digital Globe®.

Figure 2.9 illustrates the effect varying illumination and the passage of time can have. The illumination is present in the parking lot and the roof of the center building of the topmost cluster, which is a very common issue in identifying a roof. Admittedly, it is possible that this parking lot was paved and this is instead an illustration on the passage of time, this does not explain the roof. Also, note the trails that have been formed in the southwest corner of the image, that are not present in the older satellite image.

2.8.2 SIFT: Scale-Invariant Feature Transform. The Scale-Invariant Feature Transform(SIFT) is invariant to reasonable changes in scale and orientation [5]. Orientation includes both rotation of the camera (or object, depending on point of view) about the axis of the camera normal to the surface, as well as to a good extent affine transforms. SIFT is about 50% reliable at attitudes of up to 50^o, and the reliability improves the less affine the images are to each other [5]. Ideally, the aircraft witha downward-pointed camera will seldom be capturing images at 50^o, and SIFT's reliability improves with less of an affine offset. Scale has experimentally been reliable up to nearly an order of magnitude, with around an 80% repeatability rate for matching just the feature [5]. This is because SIFT down-samples an image by a factor of four (cutting each dimension in half, for one-fourth the area) several times.

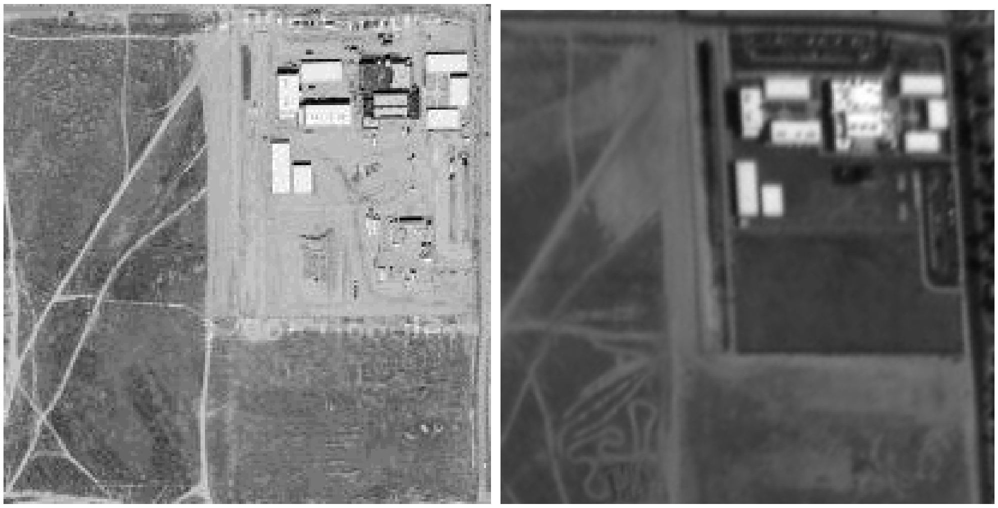


Figure 2.9: Effect of Illumination and Time Differences

In [5], a match was required to be within $\sqrt{2}$ of the correct scale. The SIFT process is performed as follows:

- **Scale-space extrema detection:** Apply the difference-of-Gaussian several times (scales) to several down-samplings (octaves), generally 4 scales per octave.
- **Keypoint localization:** Feature candidates are those which are a local minima or maxima, relative to the eight other pixel locations adjacent to it in its scale, as well as relative to the nine pixels adjacent in the scales above and below. Accordingly, it is the local maxima or minima over 27 pixels. Keypoints can be interpolated to obtain sub-pixel precision. Because of this, it is possible to have keypoints essentially on top of each other, from different octaves.
- **Orientation assignment:** Orientation and magnitude are assigned to the feature. This can be used to help filter results.
- **Keypoint descriptor:** The gradients are then analyzed as a 16 x 16 array, where each 4 x 4 subsection of the array is 'binned' into a keypoint descriptor,

which records information about the magnitude of the gradients in that bin in each of 8 orientations. Each sample is smoothed using triliniear interpolation to reduce boundary affects for bins, to avoid abrupt changes.

The number of keypoints varies from image to image, but a 500 x 500 pixel image will typically produce about 2000 stable features [5].

2.8.3 SURF: Speeded-Up Robust Features. Speeded-Up Robust Features (SURF) was developed in the wake of SIFT and other detectors that sought to improve the distinctness or improve the computation time (or both). Like SIFT, SURF focuses on invariance between scale and rotation. The authors assume "[s]kew, anisotropic scaling, and perspective effects are . . . second-order effects that are covered to some degree by the robustness of the descriptor." Like SIFT, SURF does not consider color.

The SURF detector-descriptor utilizes integral images and an approximation of the Hessian matrix. SURF implements integral images by summing all pixels in the rectangular region defined by the point and the image origin.

SURF utilizes box filters to approximate second order Gaussian derivatives, which is more of a simplification than the Gaussian filters utilized by SIFT. Because of the nature of the box filters, the image does not need to be successively smoothed, but the size of the filter simply needs to be adjusted. The size of the filter doubles in dimensionality for each new octave.

Using Haar wavelets, an orientation is assigned to each interest point in a circular neighboorhood around the interest point. By summing the responses in a $\frac{\pi}{3}$ angle, the dominant orientation is estimated. Then, at each point, a descriptor is created. A square region is created surrounding the keypoint. This is subdivided into 4 x 4 square subregions to maintain spatial information. For each subregion, a vector of four values describing polarity of the intensity changes - that is, an intensity gradient of sorts - is computed. This results in a descriptor of length 4 for each of 4 x 4 subregions, for a total of 64. The authors experimented with adding more features, higher order terms, principal components analysis, and other methods. They found that the proposed set of descriptor terms performed the best. Shorter descriptors were faster but less robust, and longer descriptors were not only less robust in experimental tests, but took longer. The one potential improvement was to consider the sign of each component of the decriptor, rather than just the magnitude. This gives a descriptor of length 128 and is more distinctive but slower; this is called SURF-128 [3]. A key merit of SURF is that the descriptors are illumination-invariant and, by virtue of the descriptor being a unit vector, invariant to contrast (scale factor) as well.

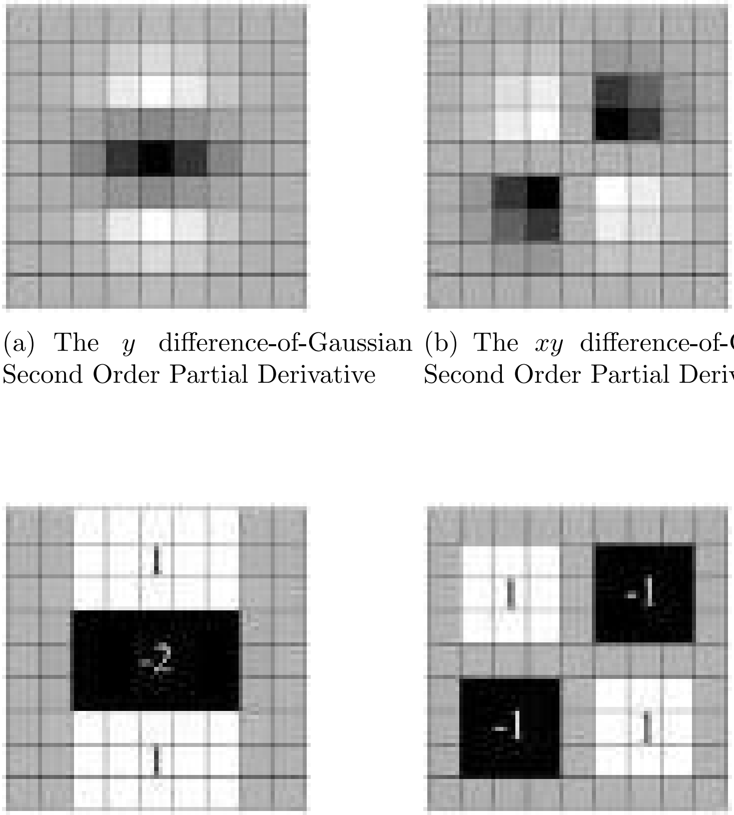
Figure 2.10 shows the filters used in SIFT and the SURF equivalents. The medium grey areas represent zero, while shades of grey closer to white and black are greater than and less than zero, respectively.

2.8.4 RANSAC. The Random Sample Consensus (RANSAC) algorithm is a random, non-deterministic process. RANSAC attempts to fit a transformation model to a random subset of all feature matches between two images. It then evaluates the developed model against all other points. It is shown in the appendices that the set of features that could be assigned deterministically as the best are not necessarily so, and by taking attempting several possibile subsets, a superior solution will be found.

The objective of RANSAC is to fit a model to experimental data. RANSAC selects a small sample of the data, develops a model, and attempts to iteratively improve the solution. This process of selecting a small set from the original set and iterating then repeats a pre-determined number of times in an attempt to generate a correct model. As the model improves, other data consistent with the proposed model are added to the solution set [4].

2.9 Prior Art

This section examines recent work in geolocation. Two works reviewed are satellite-based and discusses some of the challenges in geolocation. The third discusses exploitation of the imaging sensor to provide an attitude update. Finally, the impact of these works on this research is explained.



(a) The y difference-of-Gaussian Second Order Partial Derivative

(b) The xy difference-of-Gaussian Second Order Partial Derivative

(c) The SURF Approximation of the y Filter

(d) The SURF Approximation of the xy Filter

Figure 2.10: The filter used by SURF is an approximation of the one utilized by SIFT [3].

2.9.1 NASA's AGSI INR System.

NASA has a satellite module known as the Advanced Geosynchronous Studies Imager (AGSI), which is an Image Navigation and Registration (INR) System [9]. AGSI's mission uses the image data collected by its cameras. This image data is distorted and corrupted from the truth position by the sensor operation, satellite orbit and attitude, as well as atmospheric and terrain effects. AGSI removes distortion via two corrections, both of which could be applied to a UAV surveillance platform: a systematic correction and a precision correction.

The systematic correction is applied to acquired images to adjust for known issues with the sensor platform itself - the sensor attitudes, characteristics, and any terrain models. By applying these corrections, accuracy is achieved within a few pixels.

The precision pixel correction focuses on features to refine the solution to sub-pixel precision. Used in the correction is information from landmarks, stars (when the satellite scans passed the edge of the earth), and range data from ground control stations. Of these updates, the landmark updates are particularly important and are utilized in both image-to-map and image-to-image comparisons. Updates were done infrequently, and when they were done, only a few samples were taken.

Image registration was performed by AGSI using three techniques: landmark registration to help correct the navigation data, swath correlation for remapping and mosaicing, and coregistration to keep track of discrepancies between channels. The AGSI project had access to 18 channels of data.

This study investigated four image registration algorithms, ultimately selecting edge detection and wavelets as being ideal for this application. Image Processing itself is a four step process.

- *Preprocessing* of images removes clouds and masks a region of interest, such as a coastline, lake, or island.
- *Feature Extraction* finds control points such as edges, regions or region centers (notably, lakes and islands), contours, or wavelet coefficients.

- *Feature Matching* is performed, which involves use of a spatial transformation, a search algorithm, and a metric of similarity. All three of these can be done in various ways and are not dependent on which technique is selected for another component.
- *Remapping and Resampling* are done as needed.

Next, landmarks are registered using a three step process, which involves preprocessing, wavelet decomposition, and registration. Of note, the wavelet decomposition is iterative, but only identified regions of interest were iterated.

Their paper claimed 0.25 - 0.5 pixel accuracy could be achieved. The 3σ requirements were on the order of μrad.

2.9.2 Prediction-Based Registration: An Automated Multi-INT Registration Algorithm. This Prediction-Based Registration (PBR) [11] algorithm was designed to perform automatic georegistration of electro-optical (EO) and Synthetic Aperture Radar (SAR) imagery intelligence (IMINT). This work was sponsored by Air Force Research Laboratory (AFRL) Sensor Directorate.

Like with AGSI [9], PBR performs two types of registrations, relative (image to image) and absolute (image to map), which is done with single ray back projection. These are solved simultaneously using resectioning and triangulation, called *multiple image geopositioning* (MIG). This approach requires the use of reference imagery or multiple mission images.

The PBR uses several models to analyze imagery: a scene model, sensor model, and phenomenology model.

- The *scene model* is comprised of four submodels.
  - The *solar model* identifies the location of the sun, the time of day, and the location on earth.

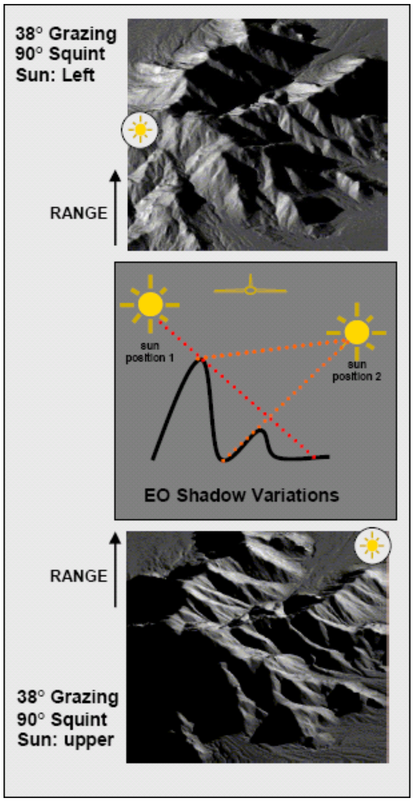


Figure 2.11: Scene Modeling with Predictive Based Rendering [11].

  - The *geometry model* describes the physical structure of the contents of the image. This identifies potentially reflective surfaces for the solar model as well as how the image will project when taken from other directions.
  - The *reflectivity model* attempts to identify the reflective potential of surfaces.
  - The *feature model* identifies unique features or control points in the image.
- The *sensor model* contains calibrations.
- The *phenomenology model* attempts to apply the reflectivity and feature models to interpret interactions of the sensors with the scene (i.e., glare).

Figure 2.11 illustrates how vastly different a scene can appear depending on different relative sun positions. The center part of the figure describes the data made available by the Synthetic EO imagery.

The PBR process is comprised of five components. First is the *input* from the database of the area being surveyed, which is then used for *scene model extraction.* This information is used to make a *prediction* of what the mission image will look like, which is then used with *image registration.* Finally, all of this is compiled into *output*, which is a synthetic reference image and a relationship from that image to the mission image and scene model.

One important component of PBR is that it handles terrain in varying lighting conditions, as shown in Figure 2.11.

Lastly, the matching algorithm is discussed. It uses a translation only technique, which is suitable for a satellite, but not necessarily so for a UAV. One alternative algorithm under consideration in the paper was the TWo-axis Image Sorting Technique (TWIST). The benefit in this case is the quick computation time.

2.9.3 Tightly-Coupled INS, GPS, and Imaging Sensors for Precision Geolocation.

An image-based method of improving the navigation solution of a UAV when coupled with GPS and inertial data is presented in [14]. The motivation behind the paper was to exploit sensors that exist in many UAVs to improve the navigation solution. The goal is to, in turn, improve the geolocation solution. The primary methodology of doing this is to improve the heading information based on features in sequential images.

This paper began by reviewing significant research in the field. Of note for this project, three approaches have been developed to improve the georegistration performance of small UAVs. The simplest was to improve the sensors on board until it meets constraints. This carries a substantial cost, in terms of size, weight, and money. The second approach is to incorporate data from previously surveyed reference targets, which has the ability to eliminate inertial sensor drift. The final approach, which is covered extensively in this paper, is to utilize image registration software.

The navigation component is handled with an Extended Kalman filter. It maintains information about both the INS and the feature tracker components. It feeds back corrections to the INS and feature locations and in turn receives measurement updates from GPS, the feature tracker, and the INS.

For each image, features are tracked automatically by utilizing the scale-invariant feature transform (SIFT). These location of these features in the subsequent image is predicted, the feature is matched in both images, and the change in location is used to correct the heading.

The paper also performed a sensitivity analysis on geolocation error. The horizontal location of a target is given by Equation (2.17). By assuming independence among each variable, the variance is found by Equation (2.18).

$$y_h = p_h + \frac{h_v}{tan(\theta)} \tag{2.17}$$

$$\sigma^2_{y_h} = \sigma^2_{p_h} + \frac{1}{tan^2(\bar{\theta})}\sigma^2_{h_v} + \frac{\bar{h}^2_v}{sin^4(\bar{\theta})}\sigma^2_\theta \tag{2.18}$$

In the above equations, y_h is the horizontal location, p_h is the location of the UAV, h_v is the projected distance, and θ is the depression angle from horizontal to the target. The nominal state value for θ is indicated by $\bar{\theta}$.

The results of [14] was a substantial improvement to the heading, with a 1σ value of approximately $5mrad$.

2.9.4 Incorporation of Prior Work. These works provide guidance for this research as well as suggestions for future directions.

Like the AGSI system, this research utilizes spatial transformations to relate one image to another. In the case of this research, the search algorithm is non-deterministic, though the metric of similarity is deterministic. Of the two feature detectors and matchers utilized, one (SIFT) is iterative, while the other (SURF) is not.

Only a few of the models incorporated into PBR are used in this research. Of the four scene sub-models, only the feature model is implemented here due to the limitations of the data set available. Like PBR, the research has calibration information that is utilized to improve the navigation solution and geolocation accuracy, in the form of a sensor model. Use of an algorithm such as TWIST was unnecessary because the database images were pre-processed (likely in a manner similar to this), and so alternative matching techniques were used.

This research is direct continuation of the work in [14] [7]. It builds on the system described with the goal of reducing the overall covariance.

2.10 Summary

This chapter has presented three types of sensors and a set of coordinate systems to relate the measurements made by each into useful terms. It has set forth a nonlinear estimation tool, the Extended Kalman Filter, to integrate these meauserements. Then, the principle of geolocation was explained, along with a way to identify and describe features in the images being used for geolocation. Lastly, it presented some more recent works in image registration and geolocation, including the groundwork for this research. The next chapter presents the theory behind the proposed georegistered image update.

III. Methodology

This chapter presents a detailed description of the method used to conduct this research. It begins with a list of all acting assumptions and then proceed to an overview of the whole process. After the overview, the significant components are detailed further.

3.1 Process Overview

The objective of this research is to enhance the precision of navigation for a UAV. It is conducted in three parts. The methodology chosen to accomplish this was to develop an additional update for the navigation system of the agent performing the geolocation so it will have a greater accuracy and certainty.

First, a method is developed that will allow an aircraft to determine its absolute position and orientation based on georegistered map data. To accomplish this, features (such as a road junction or the corner of a building) needed to be successfully matched from the camera on the aircraft to the map data. Based on the location of these features in the camera, the position and attitude of the aircraft can be estimated. The process is depicted in Figure 3.1. Several sub-studies were conducted, including:

- transforming (or not transforming) the images to be in the same perspective
- analyzing characteristics of features and matches to assign a probability that any particular feature or matches is correct
- stochastically matching features

(This work focuses on an aircraft as the satellite images used. It is feasible to get a view of land in an aircraft approximately proportional to good satellite imagery. While this concept would work with a land-based vehicle or an ocean vessel, neither have a good birds-eye view of the earth, so a different type of map would need to be created.)

Second, the method developed in the first part is improved upon by taking advantage of the algorithm utilized. The particular algorithm developed attempts to determine the aircraft's position and attitude many times, but only guesses which single solution is best. The adaptation made at this point is to utilize all of the guesses that are within the realm of possibility and statistically combine them into a result that is theoretically superior. This method can be described the same as the first part, using Figure 3.1.

Last, the second method (which is the statistical recombination of the first) is incorporated into an existing image-aided navigation filter. The objective is to determine whether the filter's performance improves or not over the average of multiple flights.

All three will be conducted as Monte Carlo simulations, where many simulations are done with different random noise applied to the sensors throughout.

3.1.1 Assumptions. The agent is assumed to have a functioning navigation system. A GPS is assumed to be active in that system. The system is also assumed to have an IMU to assist in attitude estimation. At least one camera, pointed down, is expected as well.

Additionally, it is necessary to know the lever arms for each of these devices relative to a common point as well as their relative orientations.

Also required is a database of registered images with an appropriate level of detail, updated sufficiently recently that the images will be recognizeable. This does not need to be georegistered data to perform landmark tracking, but for airborne, outdoor agents, the data should be georegistered.

3.1.2 Navigation with Georegistered Imagery. Figure 3.1 shows an overview of the geolocation process. This is a generalized and overview because it is conceivably possible to parallelize several of the processes that are shown in a serial fashion.

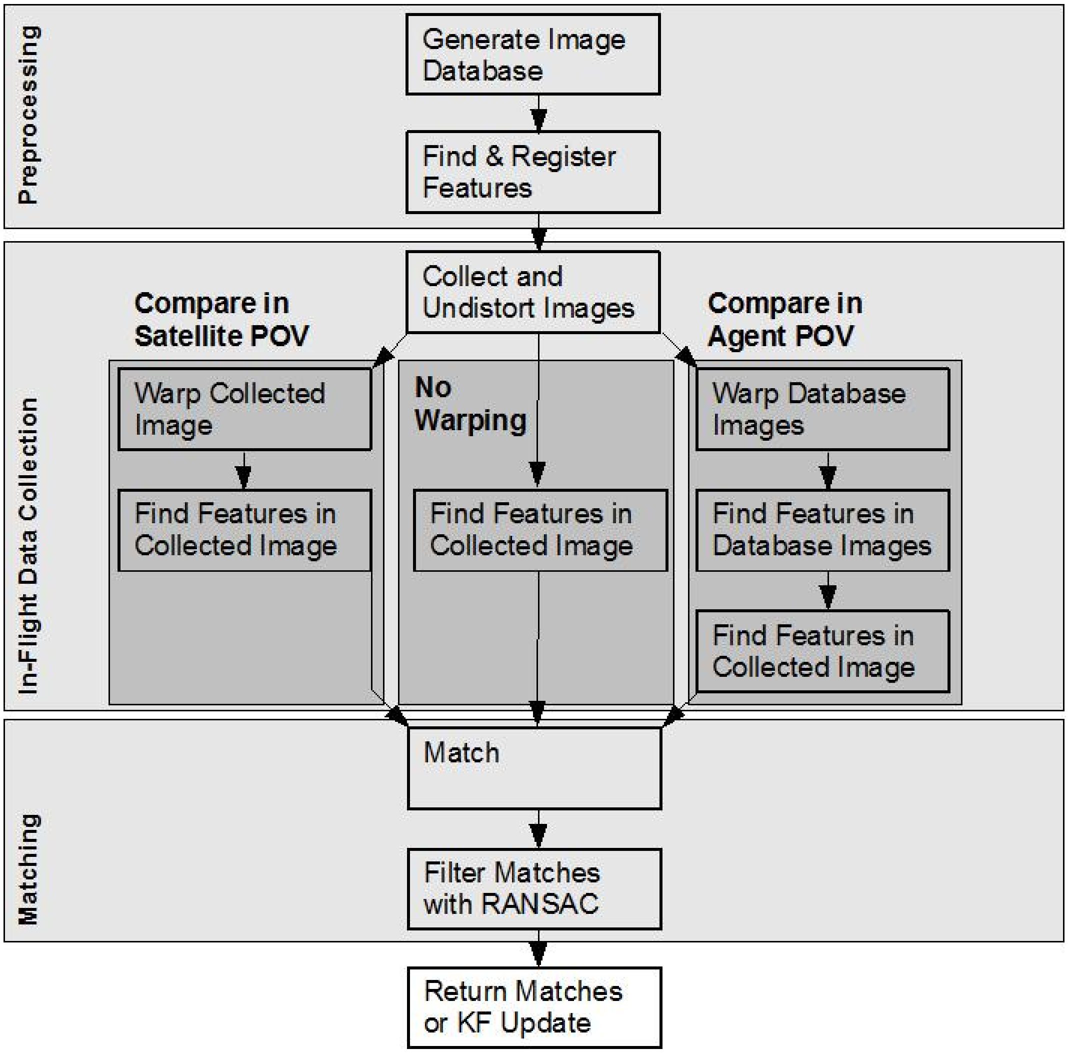

Preprocessing
Generate Image Database
Find & Register Features
In-Flight Data Collection
Collect and Undistort Images
Compare in Satellite POV
Warp Collected Image
Find Features in Collected Image
No Warping
Find Features in Collected Image
Compare in Agent POV
Warp Database Images
Find Features in Database Images
Find Features in Collected Image
Matching
Match
Filter Matches with RANSAC
Return Matches or KF Update

Figure 3.1: The above flow chart conveys the process for performing an update using registered imagery. The rest of the chapter is dedicated to explaining the process.

For this research, the truth data was collected with a military GPS receiver coupled with a tactical grade inertial measurement unit (see Section 2.3) aboard an aircraft. The high grade sensors yield what is treated as truth data.

3.2 Preprocessing

One of the significant assumptions in this project is that the agent will be operating in a previously surveyed environment. This is because all features detected will be compared against the feature database in an attempt to assign a physical location to each. This is essential to the update process. Accordingly, the feature database must be generated. Because the flight being simulated utilizes real flight data, it is sensible to utilize satellite imagery, which is readily available.

The database is implemented in tiles, each roughly 640 x 610 pixels in size, representing about 2 km x 2 km segments of land. This will require the following actions, all of which can be done prior to take-off:

- **Image collection:** minimally distorted imagery must be recorded. For this research, satellite imagery made publically available by Google Maps was utilized. A specific tile center, zoom, and pixel size had to be specified to request an image. Every tile requested overlapped with all adjacent tiles by 20-30%.
- **Feature detection:** each image is processed using the feature detector, in this case twice, once each by SURF and SIFT. The result for each tile, a pixel location (X,Y) is be assigned. The origin of an image tile (1,1) is the topmost, leftmost pixel in the tile. Then, a descriptor will be generated for each feature, along with any other algorithm-specific data.
- **Registration:** The location of each feature, which has been identified in each tile with sub-pixel precision, must be assigned a latitude and longitude. This was calculated by directly requesting from Google maps, which returns the latitude and longitude when given the feature locations and respective tile specification. The features can be specified with sub-pixel precision.

3.3 Image Processing

Once the database has been populated, the agent is ready for operation. While it is operating, images taken by the agent will be analyzed for features, which will be compared to the database of features previously collected. In doing this, it is necessary to remove image distortion for proper estimation of pixel locations.

3.3.1 Remove Distortion.

Distortion removal is based on the calibration, which provides a generalized model for transforming from the raw image to a more accurate image. How and why this is done is explained in Section 2.4. An example of image distortion can be seen in Figures 2.7 and 2.8. With the distortion removed, the agent can proceed to detect features, or it can attempt to transform either the database image tiles or the undistorted captured image to be in the same plane.

3.3.2 Image Perspective Transformations.

Once distortion is removed, it is investigated whether it is beneficial to compare using a uniform point of view. Two points of view are possible at each epoch: the view from the agent, and the view from the satellite. Image transformations serve to re-scale the image and re-orient the image. As stated in the SIFT and SURF literature, both transforms are theoretically unaffected by these transforms, given the assumed operating conditions of this project. However, both suffer as a function of affine transformation [5] [3]. Using the benchmarks developed by [8], it is hypothesized that because this is a very substantial contribution to errors, that by removing the affine offset feature matching will be substantially improved, enough to enable reliable navigation. The hope is that applying such a transform before computing features and matching will negate some of the previously mentioned errors due to orientation, as mentioned in Section 2.8.1.2. Examples of the two points-of-view (POV) are shown in Figures 3.2 and 3.3. Matches in the agent and satellite planes are shown in Figures 3.4 and 3.5.

Costs and benefits of utilizing transforms prior to computing features and matching include:

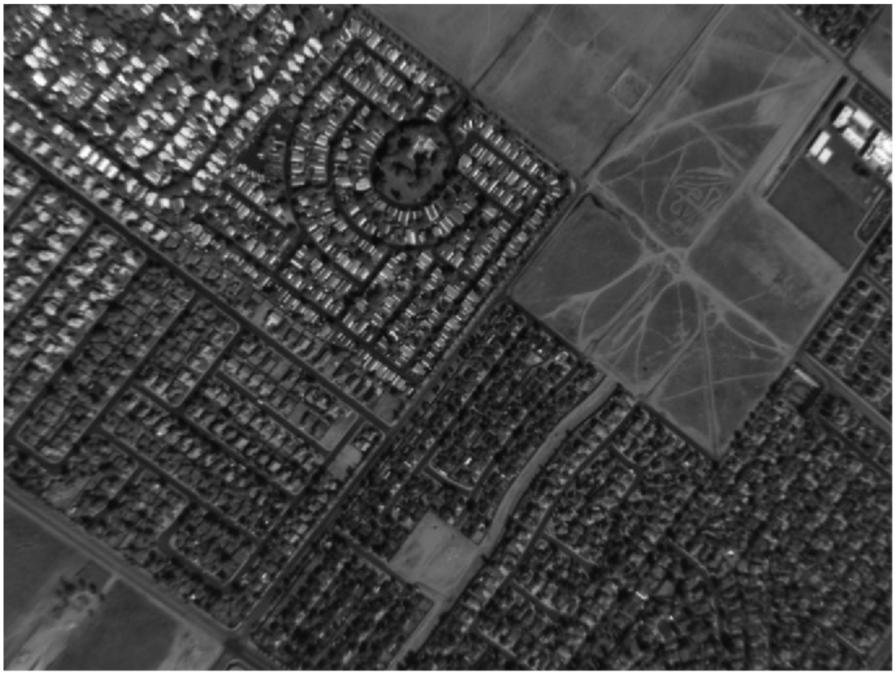


Figure 3.2: This image of Lancaster, CA was taken from the test aircraft, and thus is in the agent perspective.

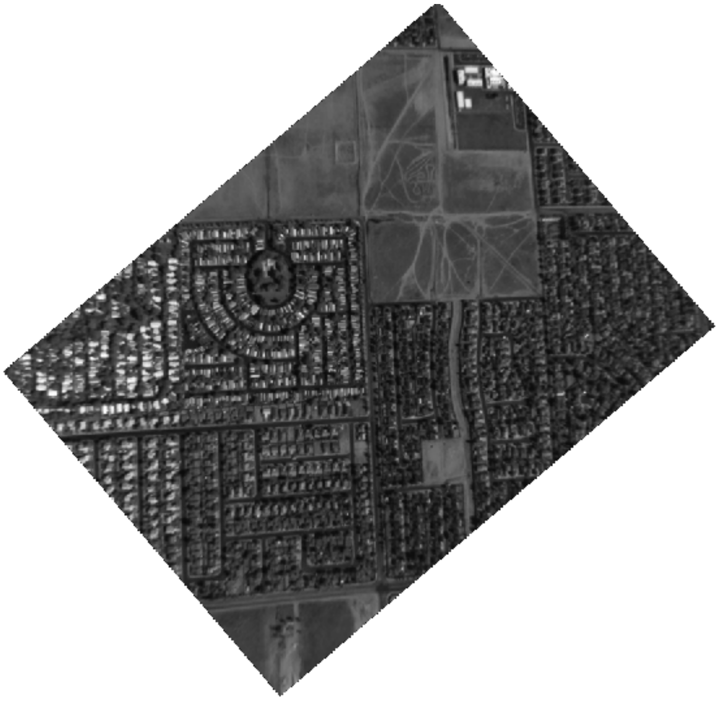


Figure 3.3: This shows the effect of transforming Figure 3.2 into the satellite point of view.

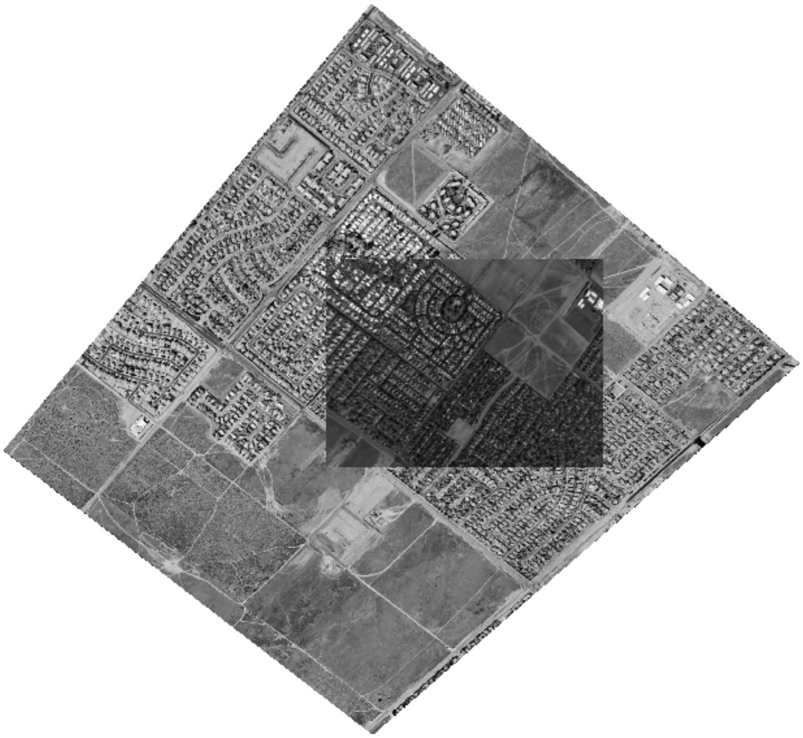


Figure 3.4: This shows an image taken by the agent overlayed on imagery from the database which has been transformed into the agent's point of view. The overlay does not fit properly because of distortion effects not covered by the camera model and because of warping of the database image relative to the agent image that is not covered by the transformation.

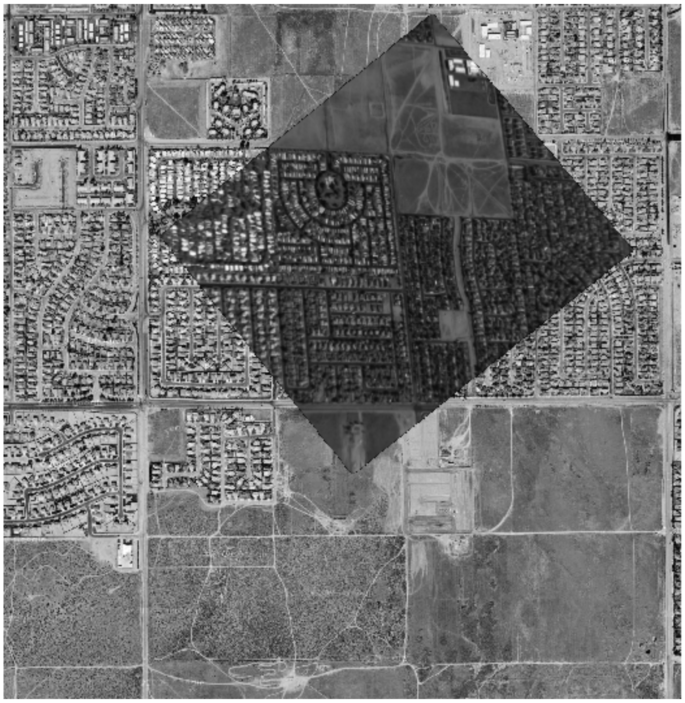


Figure 3.5: This shows a match done in the satellite point of view, with the image from the agent overlayed on top. The overlay does not fit properly because of distortion effects not covered by the camera model and because of warping of the database image relative to the agent image that is not covered by the transformation.

- **Computation cost:** By performing a transform while the agent is operating, this increases the amount of time before a result can possibly be obtained, making it less 'real-time'.
  - Comparing in the satellite plane only requires that the captured image, in this case a 780 x 1024 image, be transformed and processed with SIFT or SURF.
  - Comparing in the agent plane requires transformation of each incorporated database tile image, re-processing the image with SIFT or SURF, and estimating the latitude and longitude offline - rather than requesting the exact position from Google (though they are likely performing the same approximation) - and then also processing the acquired image. This easily has the potential to be approximately 400% slower than comparing in the satellite plane or by not transforming at all.
- **Benefits:** By implementing these transforms, some error reduction should be possible.
  - **Reduction in errors caused by affine transformation.** This was documented as a leading source of error for SIFT and SURF matching, especially when the viewpoint angle approached 50^o.
  - **Reduction in errors caused by scale.** SIFT's pixel selection for resizing to different octaves does not interpolate. Satellite imagery very commonly has colorations, lines or other markings that are only one pixel wide, and proper transformation with an included interpolation will reduce the impact, if any, of the loss of these marks. Lowe did recommend a pre-blurring be applied [5]. It is possible that, with the pre-blurring, the effect of the lossy downsampling will be diminished.

3.4 Matching

The objective of this research is to enhance the navigation solution of an agent with foreknowledge of its terrain. The following assumptions are made:

- The WGS-84 coordinates of the camera can be estimated via the agent's navigation system
- An estimation of the agent's attitude is available
- The relative orientation of the camera to the body frame is known, within a reasonable tolerance (on the order of a few milliradians and millimeters)
- A database of images is available with
  - an appropriate level of detail; that is, a scale factor per pixel within a factor of 8 of the agent's camera
  - a known frame of reference, called the database frame (outside of this section, it will be treated as identical to the *e*-frame)
  - a method of determining the WGS-84 coordinates of any pixel
  - a good estimate of the orientation of the image, relative to a common datum

3.4.1 Constraining.

Depending on the internal parameters utilized, SIFT and SURF are capable of identifying thousands of features. A standard test used five images of comparable size, easily yielding over 2000 points in the agent's image and another 8000 features (2000 each) over the four selected database tiles. This could potentially lead to 8000 comparisons per feature in the original image, or 16 million comparisons. Even on recent computers, this is not a negligible time investment.

Since an estimate is available from the INS as to the current position and heading and a camera calibration is available from before the flight, it only makes sense that this information would be used in an attempt to constrain matches. This was accomplished in [7], where the process for stochastically constraining the location of the feature locations is explained in detail. This method simply develops an ellipse in the plane of the image containing all points within a uncertainty. This typically would remove from consideration around 90% or more of all features in the database image. This not only significantly improves the likelihood that a correct match will be made, it also substantially reduces computation time.

The essential steps to locate and constrain a feature include:

- Image capture and removal of lens distortion
- Selecting an appropriate database tile or set of tiles of features or images
- Conversion of feature coordinates to the same frame - this can be done by:
  - Transforming the captured image to the frame used by the database
  - Transforming the database images to the camera frame
  - Applying transforms mathematically to the feature locations only
- Feature detection, description and localization of captured image and, if transformed, the database images

At this time, matching can be performed and the navigation system may be updated based on the residuals. The computation of updates from landmark residuals is presented in detail in [7]. The predicted landmark location is a function of the navigation state, feature pixel location $z(t_i)$, distance from the camera to the landmark $d(t_i)$, the camera to body direction cosine matrix C_c^b, and the camera projection matrix T_c^{pix}. So, for landmark position y in the navigation frame:

$$y^n = f\left(p^n(t_i), C_b^n(t_i), z(t_i), d(t_i), C_c^b, T_c^{pix}\right) \tag{3.1}$$

The feature location is estimated and the uncertainties are developed. The only change is that this prediction, which was previously computed in pixels, is converted into degrees to compare to the database latitiude and longitude values. The covariance contains numbers with units of pixels squared. The conversion factor Z_P^{wgs} is a function of the scale factor relating the zoom level of the agent image and the database image. The change in latitude and longitude given a change of one meter at this location is given by

$$P_{zz}^{wgs} = \left(\frac{\partial P_{wgs}}{\partial \bar{x}}\right)^2 (Z_P^{wgs})\, P_{xx}^P \tag{3.2}$$

To simplify calculations, the magnitude of the change in latitude and longitude given a change of one meter east and one meter north is utilized, computed about the center of the image. This is represented above as $\frac{\partial P_{wgs}}{\partial \bar{x}}$, since it is not a constant factor. The scale factor converts from pixels to meters. The P subscript denotes the pixel uncertainty in the image plane, and the wgs superscript denotes the covariance expressed in latitude and longitude.

In the case where the database images are transformed into the view of the agent camera, a least squares calculation is made to correspond the pixel location from the original image to a latitude and a longitude using the features stored in the database. When the database image tile is transformed to the agent's POV, the locations of the new feature are transformed into the original reference frame and then assigned a latitude and longitude. In the other cases, all data in the database consists of features with corresponding latitude and longitude values.

At this time, for each feature in the agent camera $\hat{z}^*(t_i)$, the probability is computed from its predicted latitude and longitude to each feature in the database. This is the application of the developed statistical weight P_{zz}^{wgs}. This is represented as

$$D_n(t_i + 1) = [z_n^*(t_{i+1}) - \hat{z}^*(t_{i+1})]^T \, P_{zz}^{wgs} \, [z_n^*(t_{i+1}) - \hat{z}^*(t_{i+1})] \tag{3.3}$$

Any features in the database within the predetermined distance are now candidates for matching. For this research, a statistical distance D_n of 2σ was used to give a 95% chance while still pruning a large number of features.

3.4.2 Proposing Matches. Both SIFT and SURF represent feature descriptors as a normalized vector. A perfect match is represented by an identical vector. Each feature in the agent camera image is then compared to each feature meeting the distance constraint. To compare two features, the distance between the two descriptors is found. This is done by subtracting the proposed feature descriptor from the current descriptor, squaring the difference of each element (to handle negative values), and summing these squared values. If the actual distance was necessary, the square root could be found, but this would have no effect on the final ranking of features. So, then, the minimum distance and thus best distance is 0, while the worst distance is 4.

Based on work done in [5] [3], a strong metric for whether a feature is a match is to compare the score for the proposed match to the next best and apply a threshold to accept or reject the match. This is reported as a ratio. This was used to filter out unstable features that appeared from only one perspective. While this is a useful method, it is worth pointing out that the utility of this metric is simultaneously substantially weakened yet made possible by the stochastic constraining. The ratio metric is weakened because the second best match is quite unlikely to be within the accepted region, and so the implied reliability of the match will be falsely high. It is made possible because it precludes the liability inherent in a sufficiently large database, where there are so many features that every feature will have a next best match in excess of the threshold. This method was also demonstrated to produce substantially inferior results the more affine two images were relative to each other. Accordingly, it is desirable to consider the impact this has on the results, along with alternative metrics in matching. Since it was claimed in [5] that several other methods have been tried during matching, the alternatives will primarily serve to reject false positives.

With SIFT, it is possible to improve the match rate and reduce the computation time by requiring that a potential match have a comparable magnitude (or scale) and be within predicted bounds (i.e., a form of constraining, such as stochastic constraining). The addition of orientation caused this improvement to degrade slightly, though still being superior to not considering it at all. Magnitude was not considered in this project. SURF likewise can be sped up by considering the sign of each descriptior (SURF-128). However, each of these still allowed numerous bad matches - matches that would be off by several lots in a suburb, so quality checks were developed to filter the results of matching. The primary instrument in this was RANSAC.

3.4.3 Tuning RANSAC. RANSAC attempts to develop a transform to correlate the matches of features from the agent to the features in the database tiles. The model developed by RANSAC is used to filter out bad feature matches prior to the matches being utilized in an update for the INS. It accomplishes this by selecting four points, determining a transform that will convert the four points from one image to the other, and then checks to see which other points fit this model. Once the number of points increases, a least-squares method of creating a transform from one image to another was used to improve the likelihood that the next iteration would find correct points [12].

When RANSAC determines the best model, a metric must be used to claim that one model is superior to another. The default metric is the number of matches made. However, this does not necessarily serve the desired purpose, because by contorting one image enough, bad features can and often do line up very well, indicating that the selected matches are far better than reality. To address this, two steps were taken:

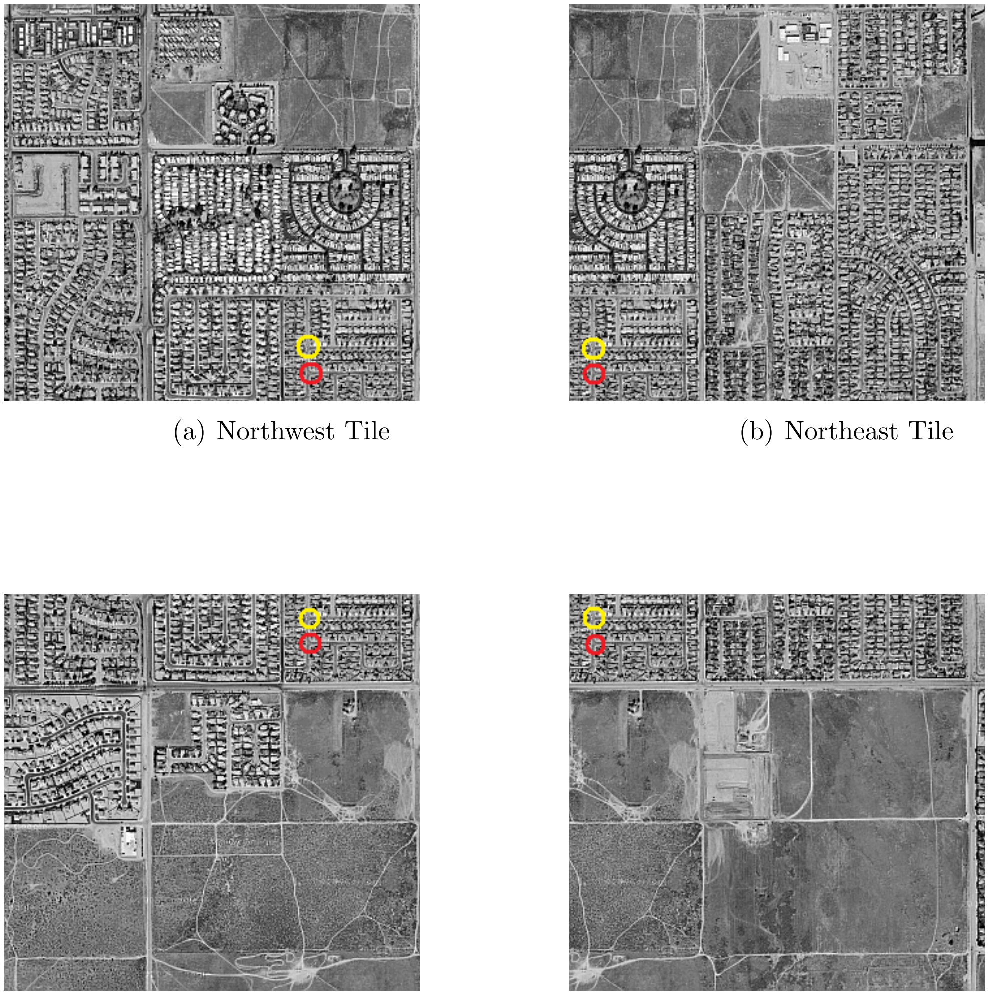
- Because the database was generated using overlapping tiles, multiple features can appear very close together. These features are often the same feature occuring in up to three other tiles. The presence of multiple features very close together gives a ‘multiple vote’ when it comes to RANSAC, increasing the odds substantially that the particular location will be chosen. Because the original metric used by RANSAC to pick the best set is the set that has the most matches, so even if all of the matches are bad, chances are good that the multiple match locations will be included because the metric increases substantially when this happens. To solve this, the feature sets from each database tile utilized are combined. Then, if any matches are within a pixel of each other, only the strongest match (the pair of features with the least distance between their descriptors) was chosen. This is illustrated in Figure 3.6.

- To enable the matching algorithms to do their best, then, a sub-study was conducted to determine which metrics for RANSAC gave the best matches. Because this topic was beyond the scope of this work, the results will be treated very briefly in the next chapter, with more supporting detail in the appendices.

3.4.4 Challenges in Matching. As was mentioned in Section 2.8, some challenges include the intrinsic camera parameters as well as time. Both issues combined and played a rather substantial part in matching. This was evident in images showing materials such as building roofs and paved roads. In the satellite databases, these would often show up as white or near-white. This is potentially due to clipping because of a longer exposure or the angle of incidence of light. Additionally, images captured by the agent, in an effort to reduce blurring, were likely exposed for much less time, and so the image on the whole is darker.

Another issue is when the database was constructed. The city featured in the majority of the flight data, Lancaster CA, has had numerous housing subdivisions constructed, industrial buildings erected, and parking lots created. Additionally, some images show the creation or disappearance of offroad paths created by driving or by flash floods. See Figure 2.9.

These challenges motivate the use of RANSAC and the methods developed in Appendix A to attempt to prune illegitimate matches. By doing this, a set of matches is established from which the position and attitude can be estimated. The quality of these estimates, when compared with the truth, are used to evaluate the results of the three tests, which are now described.



(a) Northwest Tile

(b) Northeast Tile

(c) Southwest Tile

(d) Southeast Tile

Figure 3.6: The tiles utilized in the database had an overlap of around 30%. This caused features, such as the two T junctions identified in all four tiles above, to occur four times during processing. Each has a slightly different position due to surveying error. Because they are essentially colocated, the solution will suffer from near-singularity.

3.5 Test Plan

This section describes the general test plan for the three major types of tests in this research. At the end of the section, how the results will be evaluated is explained.

All tests are done as a Monte Carlo simulation. For these tests, truth data is provided by a military GPS receiver and a high grade inertial sensor. This truth data is then corrupted using a normally distributed position error of $[5, 5, 10]m$ and $[5, 5, 20]^o$. All noise was randomly generated using a normal distribution. For the first two tests (the RANSAC single best solution and the Particle Filter test), this was done for each time step independently of all of the previous steps (i.e., error does not build up, but it resets). For the navigation system, it is applied to the truth data at each time step, and these can build up.

Using the corrupted data, the position of the aircraft is used to select database tiles and estimate the transform, if one was used. Then, matches were made using the process described in this chapter, according to the particular trial. The algorithms will be evaluated against the difference between the truth data and the position and attitude computed from final selection of matches. It accomplishes this by using the WGS-84 coordinates of the selected features in the database image in relation to their location in the agent image. Using these relations, the system attempts to estimate the error in the position and attitude of the aircraft. It does this by initially assuming it is located at ground level in the center while being oriented tangent to the ground, pointed North. Using least squares, it then estimates the necessary change in position and orientation to get this transformation. These changes are compared against truth.

3.5.1 Perspective and Feature Matching Test. The first test serves to study the effect of perspective transformations and the feature matching technique (SURF vs SIFT). This can be summarized as seen in Table 3.1. The objectives can be summarized as follows:

Table 3.1: List of Trials

Trial	POV	Detector
1	Satellite	SIFT
2	Image	SIFT
3	Both	SIFT
4	Satellite	SURF
5	Image	SURF
6	Both	SURF

- Discern which method has the highest accuracy (errors closest to zero) and the greatest precision (least variance)
- Characterize the mean and variance of the best-rated solutions

These will involve trials of processing about 100 images using each of the six methods described above. RANSAC will be used 250 times, and the single best solution will be selected each time. The single best solution is determined by a combination of factors, including the quality of the features used and number of matches. This is described in detail in Appendix A.

Every 30^{th} image from the chronologically sorted data set was selected for evaluation. The objective of this was to spatially de-correlate subsequent evaluations. Each run, truth data was corrupted as previously described. Each run had no input from the previous one, so a bad solution for one image will not preclude a good solution being found on the next.

Results are evaluated by applying each of the three bases (Satellite, Image, or Both) for comparison and both feature detectors to a set of images taken from Lancaster, CA. The results of this test are used to focuse on just one method in the next test.

3.5.2 Particle Filter Test. The objective of this test is to evaluate the quality of the solution of the algorithm when the RANSAC portion is treated as a particle filter, and multiple solutions are evaluated and then recombined, rather than simply choosing the best. This test took in all values that were developed, including solutions at great risk of singularities in the RANSAC-generated transform. This test only utilizes the best method, as evaluated in the results to the previous test. Only 100 iterations were permitted in RANSAC, because every iteration contributes to the solution, not just the best one. Because the best individual solution need not be near-perfect, fewer runs are needed.

3.5.3 Navigation System Performance Test. To implement this as an update to the navigation filter previously developed, the results of one pass through the whole algorithm with the particle filter implementation was represented as a change in position and attitude from the current state as well as the associated covariance, which was presented as a standard deviation for each term in the state. The ensemble mean and standard deviation of many runs combined will be evaluated against a previous similar test that did not have this update. The details of this update are presented in the next section.

3.6 Updating

With the objective of enhancing the navigation system on the agent, the solution developed in the last section needs to be applied to a navigation system. This section presents the method in which this was accomplished. First, the mathematical development of an update is presented. Then, an explanation of the Kalman filter integration is given.

3.6.1 Developing an Update. After performing matching, position and attitude estimates can be derived from the results. Because each match gives two constraints (the x and y coordinate of the feature in the image), at least three matches are needed to get position and attitude.

A weighted least-squares approach is used to find the error in the position and the attitude. A very useful trait of RANSAC at this point can be exploited: because multiple repetitions are performed over the whole set of features, a matching set of features is computed each time, providing multiple potential solutions. By solving each set RANSAC develops, a data set is populated with the estimated position and attitude errors. Each term is weighted inversely proportionally to its fitness. The fitness predicts the accuracy of the solution based on the characteristics of the matches selected. The concept is developed thoroughly in Appendix A.

Equation (3.5) shows the development of a rotation and translation matrix as the result of least squares. It requires first augmenting the position of the targets in the camera frame (building s^c_{aug}, in meters), as shown in Equation (3.4).

$$s^c_{aug} = \begin{bmatrix} s^c \\ 1 \end{bmatrix} \tag{3.4}$$

$$\begin{bmatrix} d\mathbf{C}^e_b & d\mathbf{P}_e \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} = \left(s^c_{aug} \cdot \mathbf{W} \cdot \left(s^c_{aug} \right)^T \right)^{-1} s^c_{aug} \cdot \mathbf{W} \cdot \begin{bmatrix} \mathbf{T}_{e,3xn} \\ \mathbf{1}_{1xn} \end{bmatrix} \tag{3.5}$$

The change in position, in the e frame, is given by $d\mathbf{P}_e$. The vector $\mathbf{W}$ is the individual fitness or confidence assigned to each individual match in the calculation, as specified in A, normalized so it has a mean of 1. The matrix $\mathbf{T}$ represents the location of each target, in the e frame, as determined by the match to the database.

The errors in attitude are given approximately in the off-diagonal terms, as shown in Equation (3.6). The values were found after normalizing the matrix to have a magnitude of 1. Per [13], it is necessary to shift into the navigation frame and use the small angle approximation to compute the change in attitude (in euler angles).

Each set computes a position offset and a rotation offset because of the relation between T_e and s^c. These offsets represent the residual between the predicted state and the measured state, and are computed as $d\mathbf{C}^e_b$ and $d\mathbf{P}_e$ in Equation (3.5). The resulting residuals are the computed error in position and attitude. However, for these updates to be useable by a Kalman filter, the covariance must be estimated. From the set of all solutions generated by RANSAC, statistics can be generated for use with the Kalman filter. The values are weighted by the overall iteration fitness given by RANSAC. The measurement error is presented as the weighted standard deviation of these estimates, using the same $\mathbf{W}$ as in Equation (3.5). This is computed as shown in Equation (3.7).

$$\begin{bmatrix} 1 & -d\psi & d\theta \\ d\psi & 1 & -d\phi \\ -d\theta & d\phi & 1 \end{bmatrix} = \frac{\mathbf{C}_b^n \cdot d\mathbf{C}_n^e}{norm\left(d\mathbf{C}_n^e\right)} \tag{3.6}$$

$$\mathbf{R} = diag\left(\sqrt{\sum_{i=1}^{N}\left(\frac{\mathbf{W}_i}{N}\left[z_i - \bar{z}\right]^2\right)}\right) \tag{3.7}$$

3.6.2 Incorporating an Update. The statistics developed above will be incorporated into an update to the EKF presented in [14]. The notation utilized in this section references Section 2.6.

The mean and standard deviation used will be the weighted standard deviation from the RANSAC repetitions. In reference to Equation (2.13), the mean represents

$$\delta x_{PF} = \begin{bmatrix} d\mathbf{P}_{e,3\times1} \\ d\phi \\ d\theta \\ d\psi \end{bmatrix} \tag{3.8}$$

In Equation (3.8), δx_{PF} represents the weighted state mean, as computed from the particle filter-style state estimate.

The Kalman gain $K(t_i)$ is computed as shown in Equation (2.12), where R is as shown in Equation (3.7). The $\mathbf{H}$ matrix is simply $\mathbf{I}_{6\times6}$, though this would change and become more complicated with a non-zero lever arm.

3.7 Summary

This chapter has laid the groundwork for performing an update based on an image captured by an agent compared geo-registered image data via various methods. The next chapter presents the results of this experiment.

IV. Data/Analysis Discussion

This chapter presents the data collected from the three different tests done. It begins by discussing the low level feature matching performed with RANSAC. Then, the results of converting individual RANSAC solutions into a particle filter-style state estimate is presented. Finally, the results of incorporating this into an existing navigation filter is shown. In each section, the data is presented in a series of figures and analyzed.

4.1 Experiment Overview

This section briefly describes the experimental setup.

As required in Section 3.4, the lever arms between each sensor are known as well as an the relative orientations. The camera is pointed straight down and is attached to the bottom of the aircraft. Data is recorded from the GPS at around 10hz, from the IMU at around 100hz, and from the camera at around 3hz. The trajectory recorded is shown in Figure 4.1. Data was collected during the day during the summer.

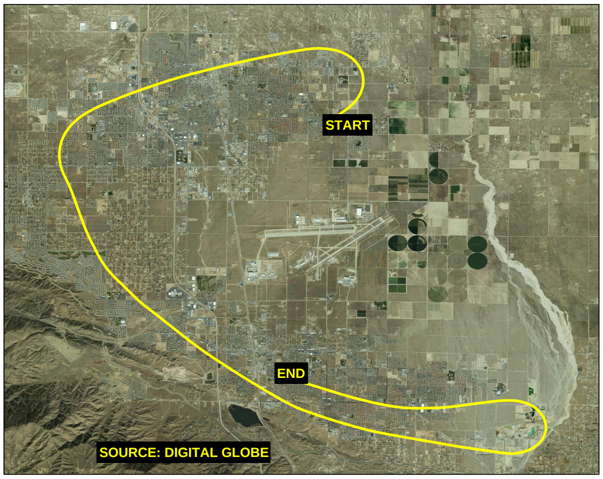


Figure 4.1: This path shows the profile of the flight taken in Lancaster and Palmdale, CA. Segments from the entire flight were utilized in the first two tests. In the final test with the integration into the navigation filter, the first 300 seconds were utilized, which came about halfway down the segment on the left. This figure was originally generated for [14].

4.2 RANSAC Single Best Solution

The test in this section has two objectives, which are restated from Section 3.5.1:

- Discern which method has the highest accuracy (errors closest to zero) and the greatest repeatability (least variance)
- Characterize the mean and variance of the best-rated solutions

The data collected for each test is presented as a collection of six plots, representing the ensemble statistics for each component of attitude and position.

While it would be possible to create a metric for determining which is best, such as selecting the one with the single best improvement or the best overall solution, none of these six tests result in a navigation solution sufficient for navigation. The worst is the case with no transformation. The case where images are matched in the satellite perspective appears to slightly outperform the case where images are matched in the agent point of view in terms of the overall solution quality. From the six figures showing the results (Figures 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7) it can be seen that the heading solution was best using SIFT in the agent point of view. The quality of the solution of the satellite-transformed cases would perform better for all other components.

Regardless, the true quality of any of these is lost in the noise. The figures shown were already the result of substantial amounts of outlier rejection. This was done because many solutions were unstable, producing nonsensical solutions that would throw off the mean. Some such solutions involved the aircraft being located several parsecs away (a parsec is 3.26 light-years). Others involved the aircraft being upside down or turned around completely. As an example of how disasterously bad some results were, see Figure 4.8.

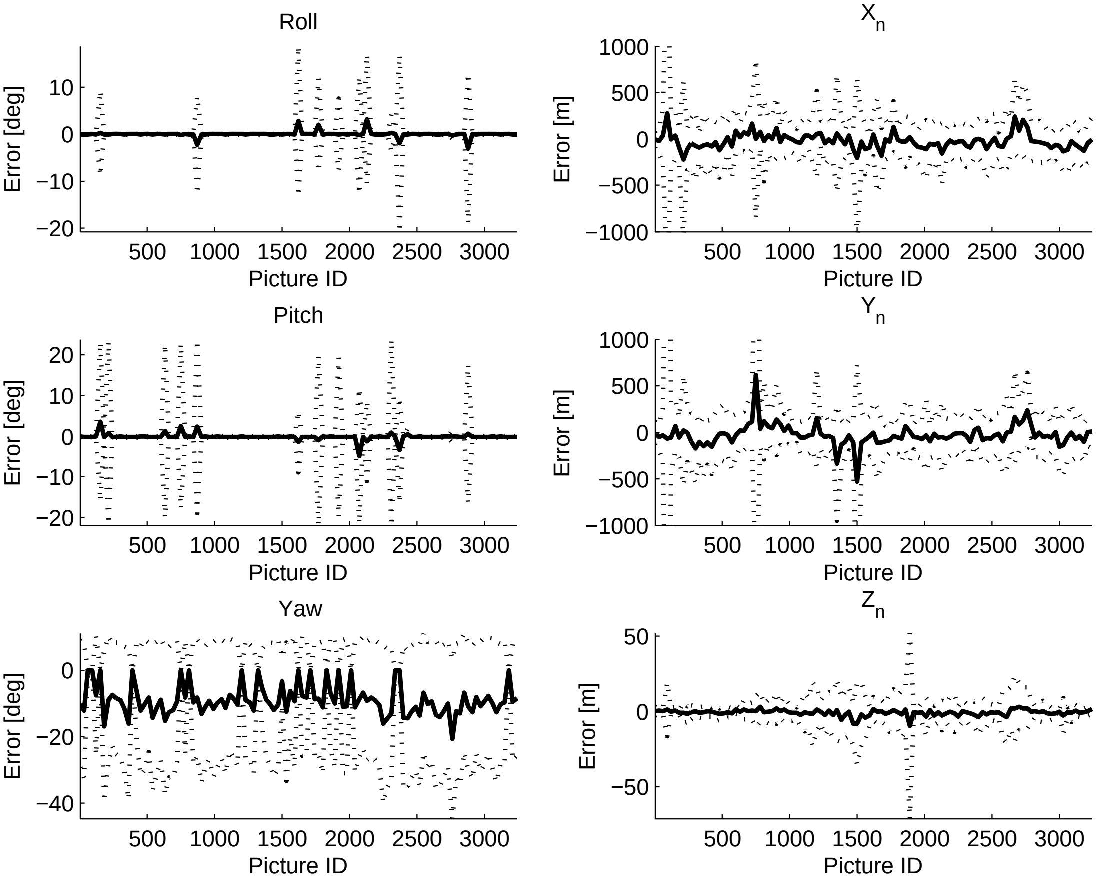


Figure 4.2: This figure shows the errors relative to truth for position and attitude using SIFT in the satellite point of view. Generally, the results were best for this case except in heading. This is possibly a product of near-singular solutions, or possibly due to challenges due to change in perspective. The dotted lines are the ensemble standard deviation, and the solid lines represent the ensemble mean.

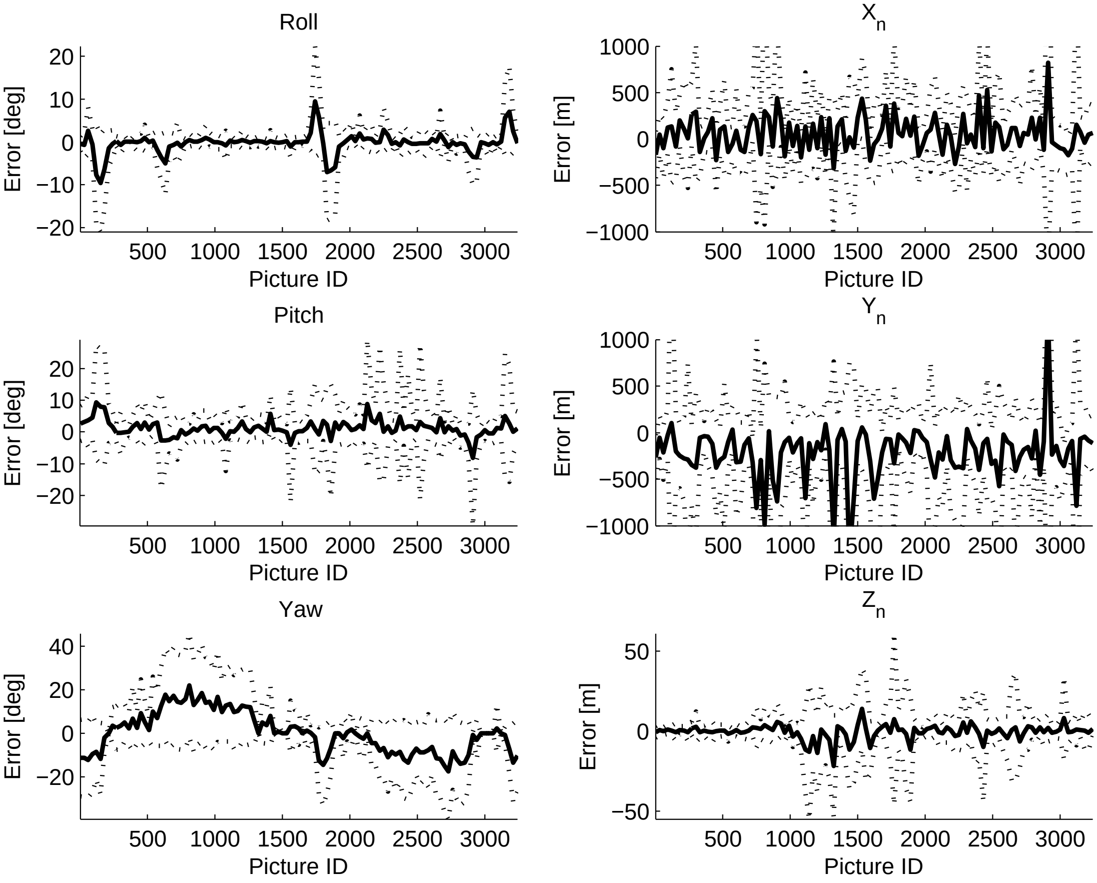


Figure 4.3: By studying the errors relative to truth for position and attitude using SIFT and without transforming either image, it is readily apparent that the lateral position and heading are both very poor. The dotted line depicts the ensemble standard deviation, and the solid lines represent the ensemble mean.

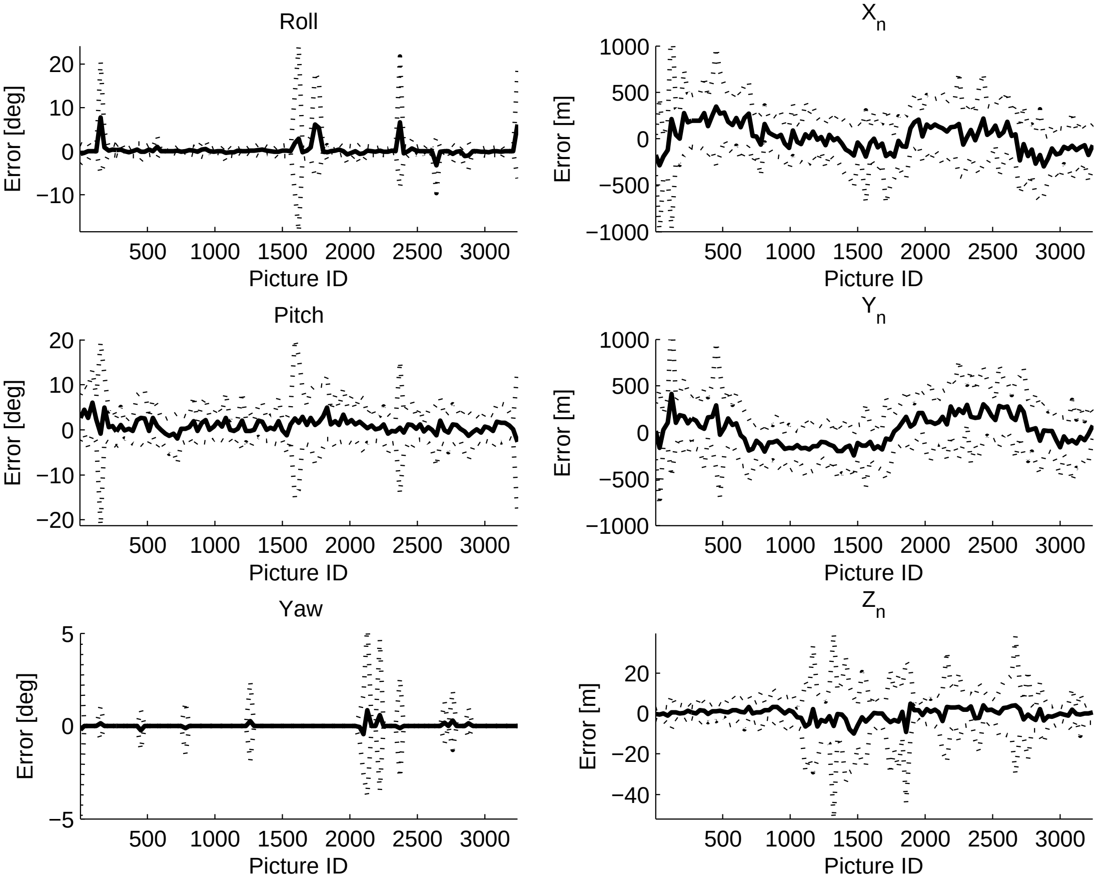


Figure 4.4: Examining the errors relative to truth for position and attitude using SIFT in the agent point of view, the most striking detail is that the heading is very accurate and precise, with only a few blips appearing in the least stable images. The roll and pitch fare worse than in the satellite point of view. The dotted lines represent the ensemble standard deviation, and the solid lines represent the ensemble mean.

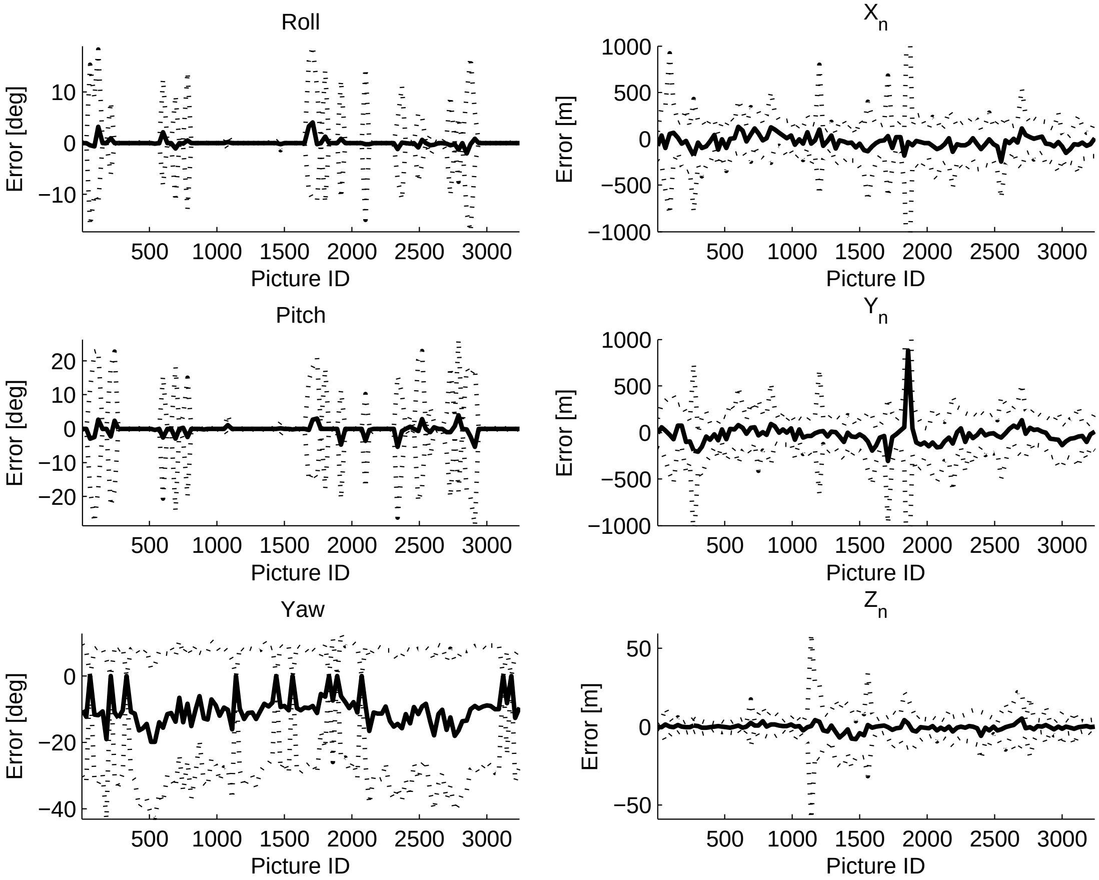


Figure 4.5: The errors relative to truth for position and attitude using SURF in the satellite point of view fare very comparably to the results achieved by SIFT. It appears that it is slightly more vulnerable to near-singular solutions than is SIFT. The dotted lines represent the ensemble standard deviation, and the solid lines represent the ensemble mean.

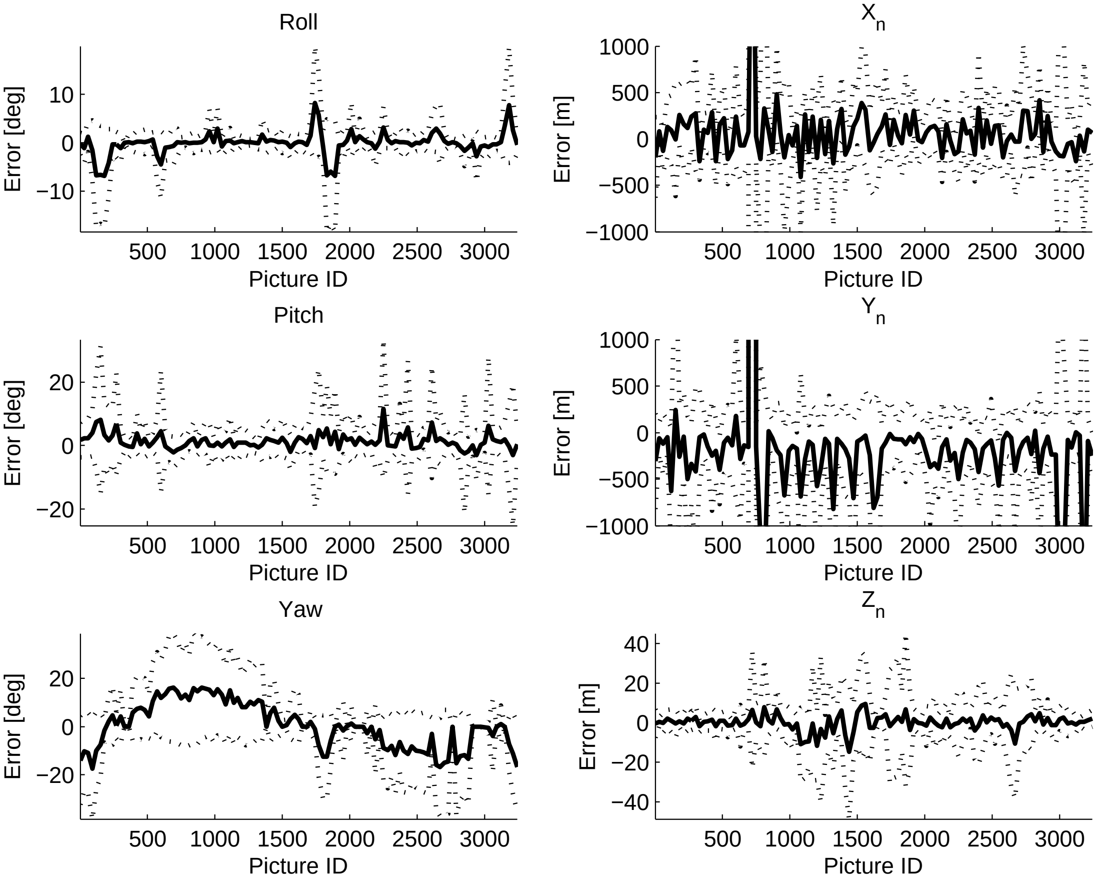


Figure 4.6: The errors relative to truth for position and attitude using SURF without transforming either image show that it is easily the least stable method. The less precise feature descriptor combined with the less precisely matching image cause an excessive number of incorrect matches, which in turn causes a large quantity of incorrect and likely unstable solutions. The dotted lines represent the ensemble standard deviation, and the solid lines represent the ensemble mean.

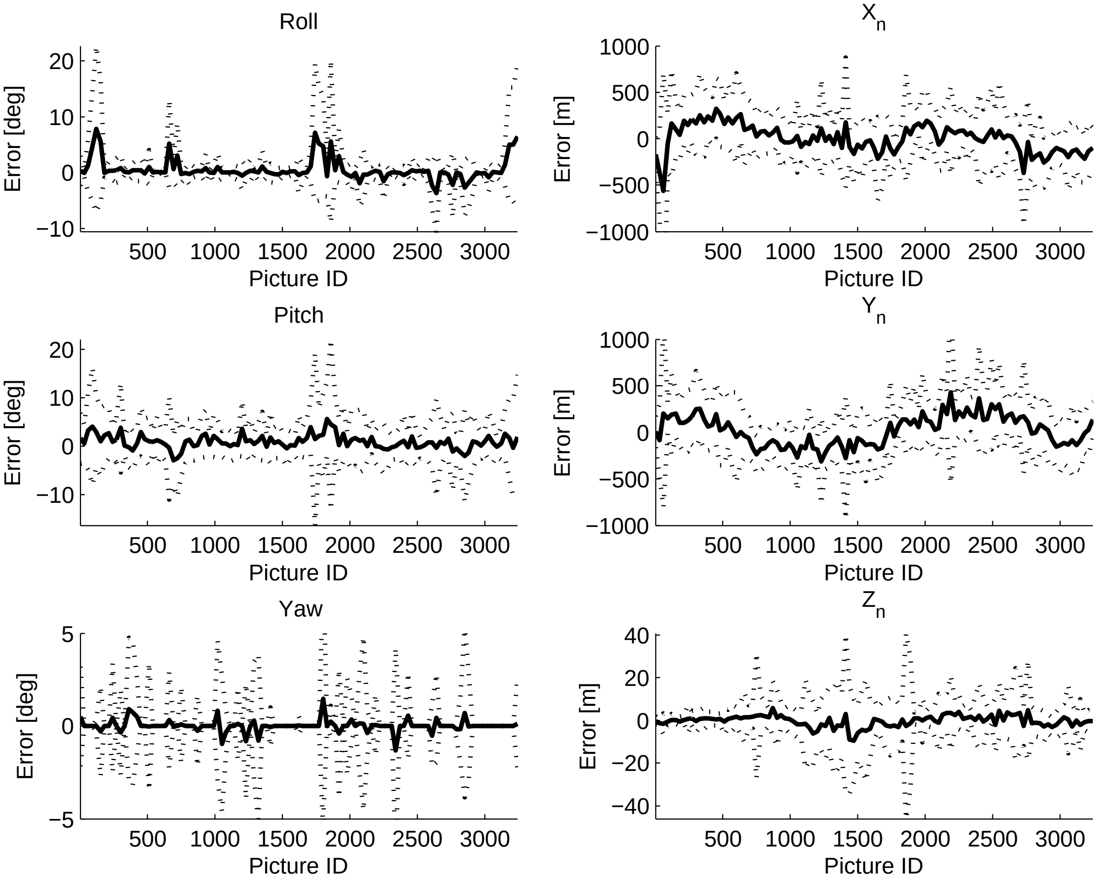


Figure 4.7: The errors relative to truth for position and attitude using SURF in the agent point of view show that, as with the SIFT solution, the heading is nearly perfect, and like all the updates the altitude is very good. However, the other four components of the state are insufficient to reliably navigate. The dotted lines represent the ensemble standard deviation, and the solid lines represent the ensemble mean.

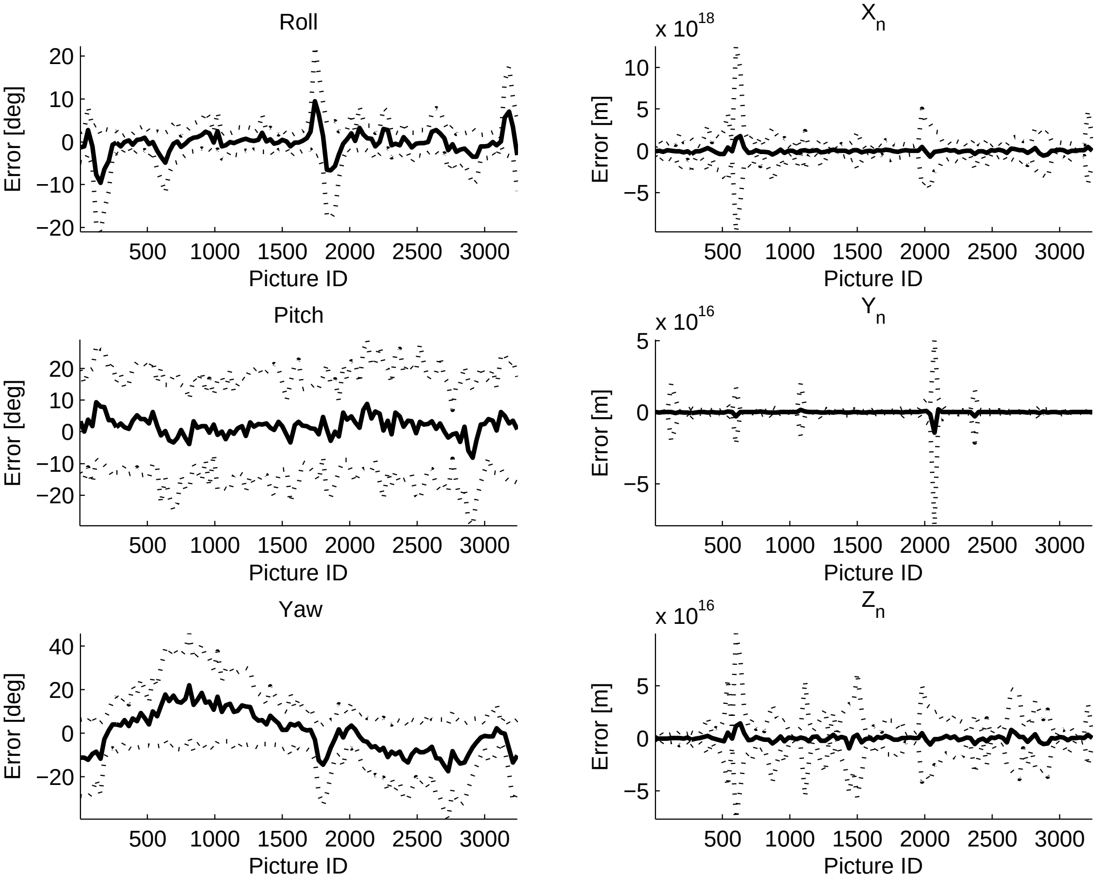


Figure 4.8: This shows errors relative to truth for position and attitude using SIFT with no transform and no outlier rejection. The position error has a tremendous standard deviation, and within 3σ, the aircraft could be pointed just about anywhere. The dotted lines represent the ensemble standard deviation, and the solid lines represent the ensemble mean.

4.3 RANSAC Particle Filter Solution

This section describes the re-combination of RANSAC solutions in a particle filter-style calculation. The result of 50 runs is shown in Figure 4.9. In general, it showed very promising results. If the singularities caused by near-colinear combinations of points was mitigated, this plot would show very excellent results across the board.

A glimpse of some of the potential benefit to be realized is presented in Figure 4.10. This figure conveys the best stretch from each component and has been rescaled to highlight the improvement. The best components of the roll exhibit a standard deviation of error around 0.001^o, which is well below the sensor noise threshold. The pitch error may have a bias included, around 0.1^0 in magnitude. Its standard deviation of error is approximately that much as well, which causes the 1σ bound to not include the truth value in a few places. The best errors east and north have uncertainties in error around $200m$, which at an altitude of about a kilometer is substantial, but if it was an update in a GPS-denied environment, it may be a borderline sufficient navigation estimate. The best vertical errors are actually in a different time frame than the other four states mentioned thus far, but it is nearly right-on and has an uncertainty around $5m$, which would be more than sufficient for navigation.

4.4 Navigation Filter Update

A Monte Carlo simulation of 18 runs was performed using the SIFT detector on data compared in the satellite point of view, and with the state estimated using the particle filter-style computation. Figures 4.11 and 4.12 show the results of the run. Each of them suffer from a glaring issue: much more than the expected 32 percent of the time, the ensemble mean is beyond the 1σ bounds. Again, the impact of singular and near-singular matrices in the least squares computation due to colinear or near-colinear features shows up in solutions that exhibit very poor accuracy (see Equation (3.5)). Additionally, because of the potential precision available, even bad updates can be given a very low covariance because the standard deviation of the

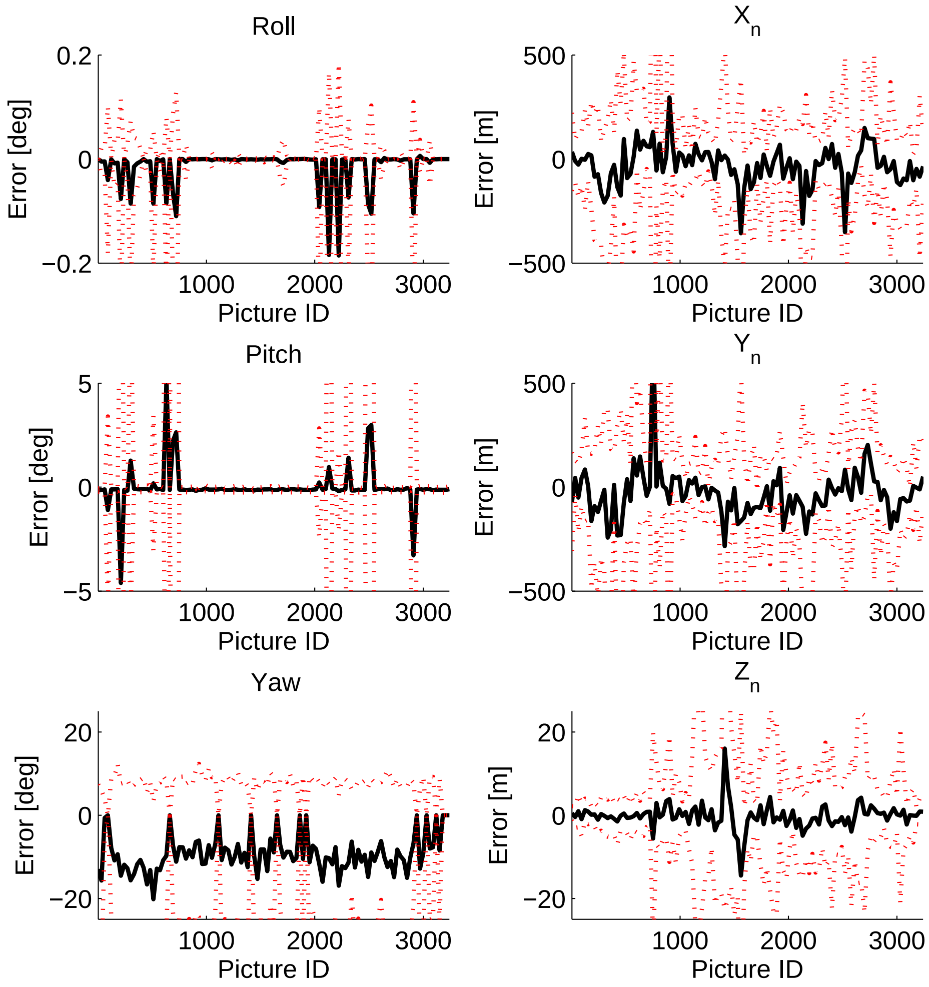


Figure 4.9: The results from 50 tests implementing a particle filter-style position and attitude estimation solution demonstrates significantly better results than the single-best solution RANSAC model, with the exception of the heading. The heading errors are due to the singularities discussed in this section, as the heading was impacted more than other states. The exceptional performance (hundreths of degrees, or tens of microradians) of the roll and pitch at most times is dwarfed by the images that had the most problems with near-singular solutions. The dotted lines are the ensemble standard deviation, and the solid lines represent the ensemble mean.

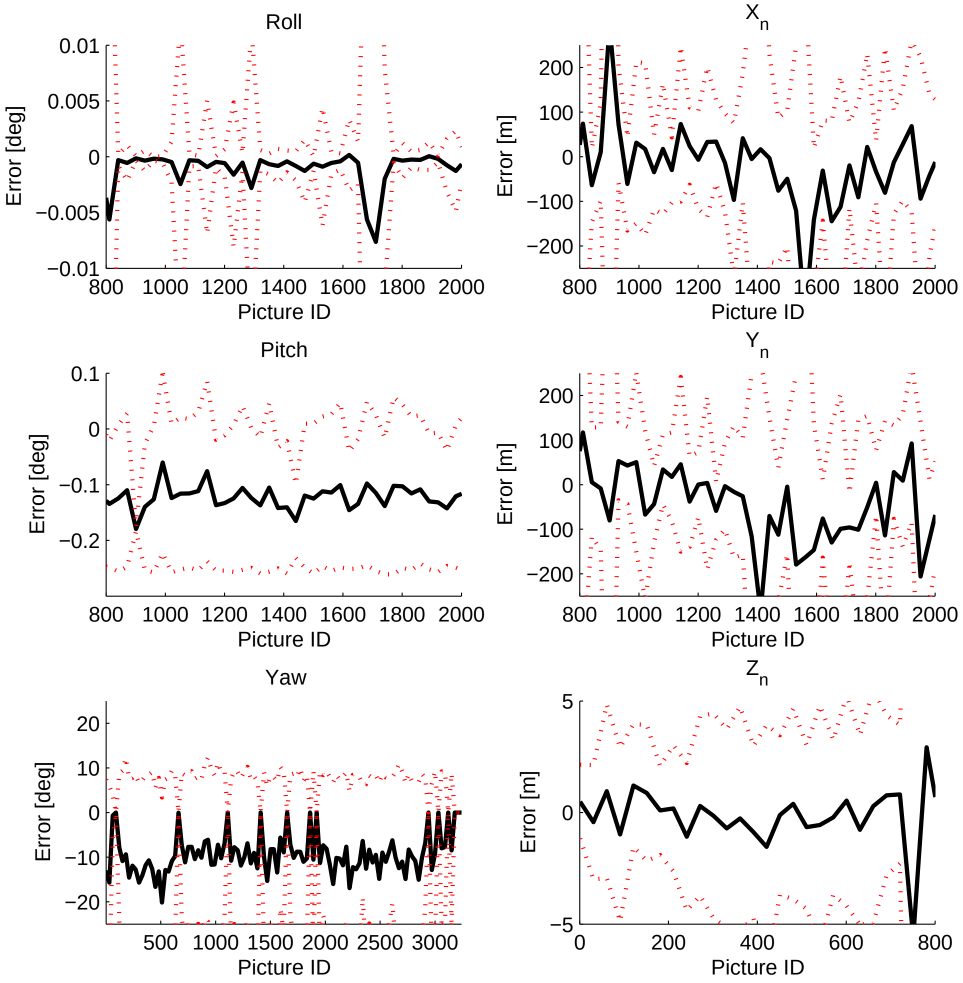


Figure 4.10: This highlights the better portions from Figure 4.9. The only portion not to have a significant portion demonstrating good performance is the heading, which is simply incorrect the entire time. From this figure, it is possible to estimate good performance for the other five states. For comparison, an error of 0.1^o is roughly $1.7mrad$. Once all the errors are accounted for, it is predicted that the error in the heading will snap to around zero (it does have a few places where it gets the heading very precisely), and the other uncertainties will improve substantially as well. The dotted lines are the ensemble standard deviation, and the solid lines represent the ensemble mean.

many residuals calculated are low (see Figure 4.10). When a bad update with a low covariance is presented to the filter, any errors present will propagate rapidly.

It should be noted that in this section, unlike the other sections, outlier rejection on the data shown in the plots is not performed; and all recorded values are included. This is done so the results presented realistically portray what would happen if this were integrated as it stands now. What is reflected is that the filter rejected any position residual (and its corresponding attitude) outside of a 50σ bound. This value was designed with the goal of bounding unreasonable solutions that are hundreds of σ or more away while allowing for an update to occur, since realistically the position solution simply will not approach the precision of a military GPS receiver.

As the results of this section are in part being compared with previous results, it is pertinent to include a basis for comparison. This is given in terms of the attitude and is shown in Figure 4.14 and can be compared easily with Figure 4.13. This data is taken from [14]. It is not useful to compare the position errors, since they are on the order of meters in [14] and on the order of tens of meters here; it is clear that the results in this case are inferior, and the position solution has been degraded by incorporating this update (note typical position solution and heading results in Figure 4.9). Additionally, [14] found that the position was negligibly affected by the incorporation of the attitude update from the camera. However, because it did affect the errors for attitude, the data is included here. It can be seen here that the net result of this update decreased the performance of the filter.

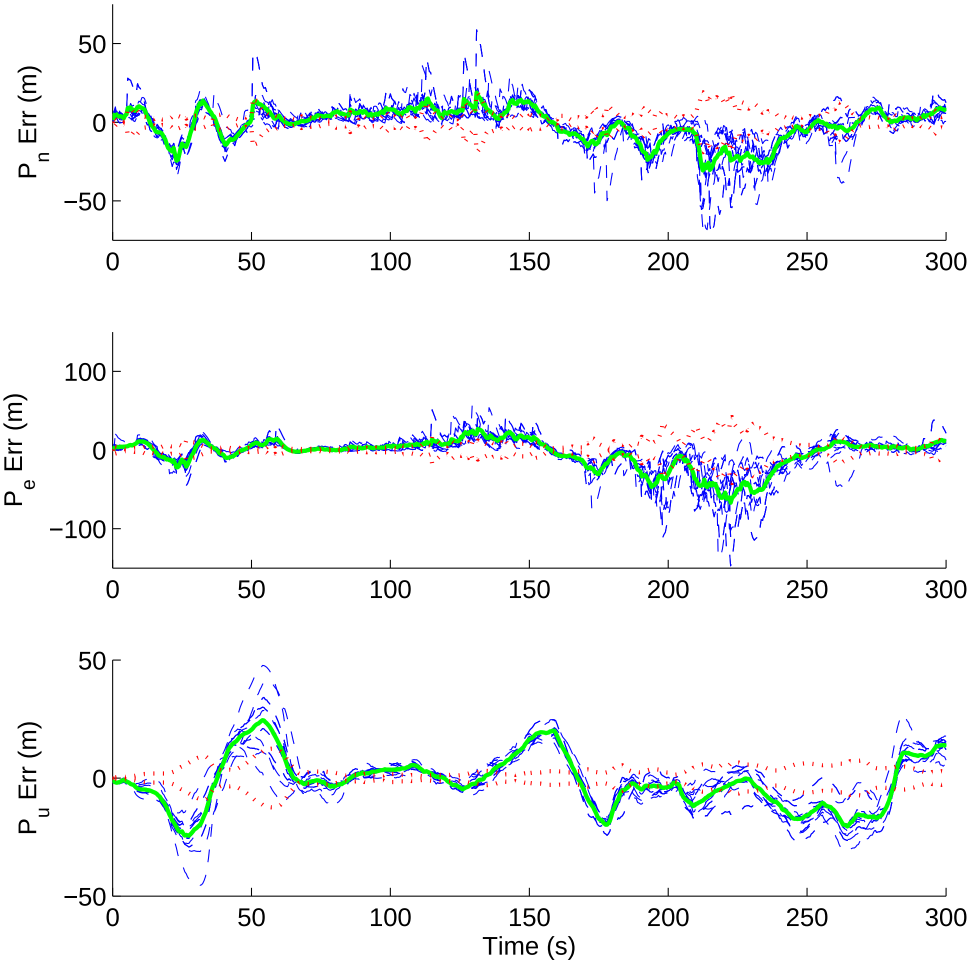


Figure 4.11: The position estimates as a result of the simulation are shown here. The many dashed lines represent an individual run. The solid line is the ensemble mean, and the two dotted lines represent the ensemble 1σ standard deviation.

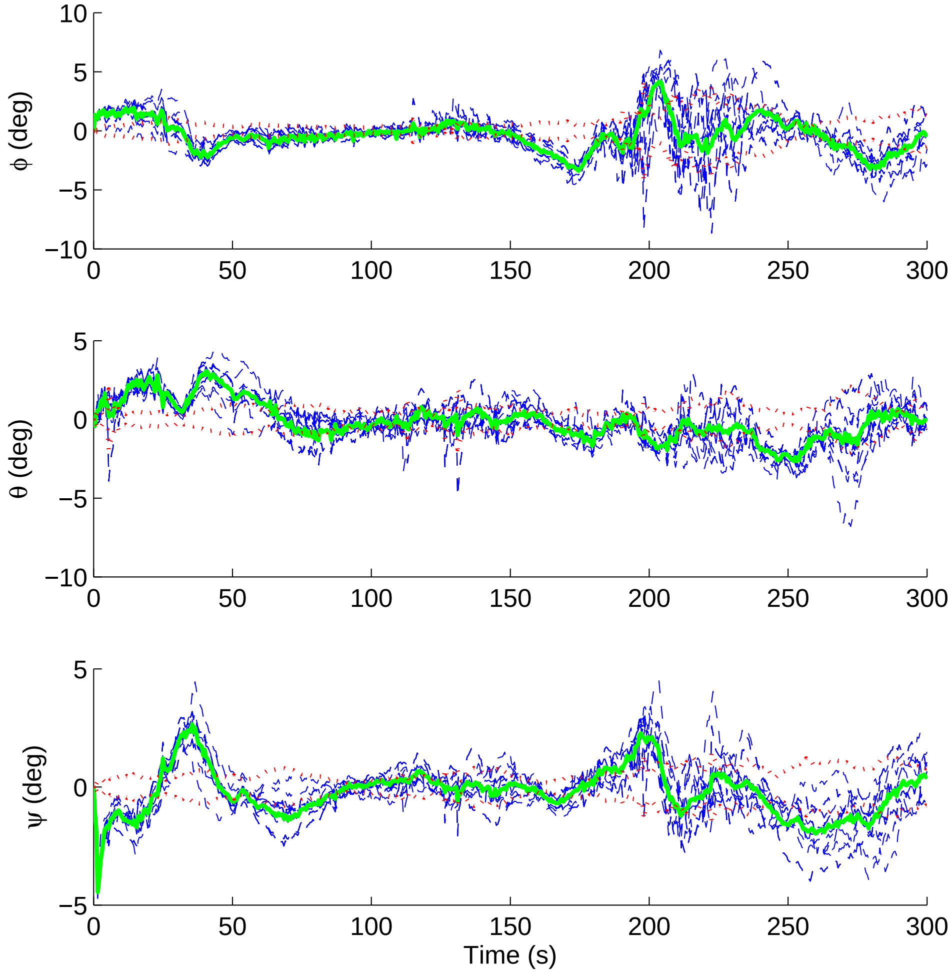


Figure 4.12: The attitude estimates as a result of the simulation are shown here. The many dotted lines represent an individual run. The solid line is the ensemble mean, and the two dashed lines represent the ensemble 1σ standard deviation.

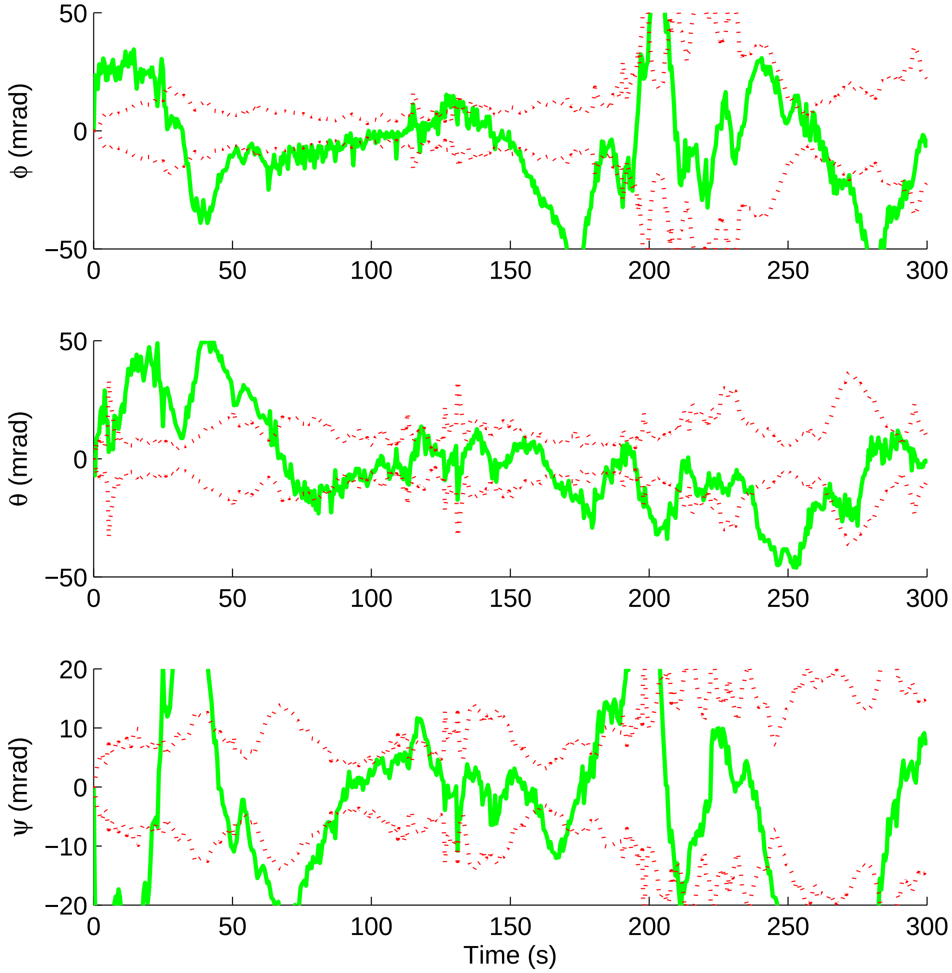


Figure 4.13: This is essentially the same as Figure 4.12, but scaled to use *mrad* as its units instead, for comparison with Figure 4.14. The solid line is the ensemble mean, and the two dashed lines represent the ensemble 1σ standard deviation.

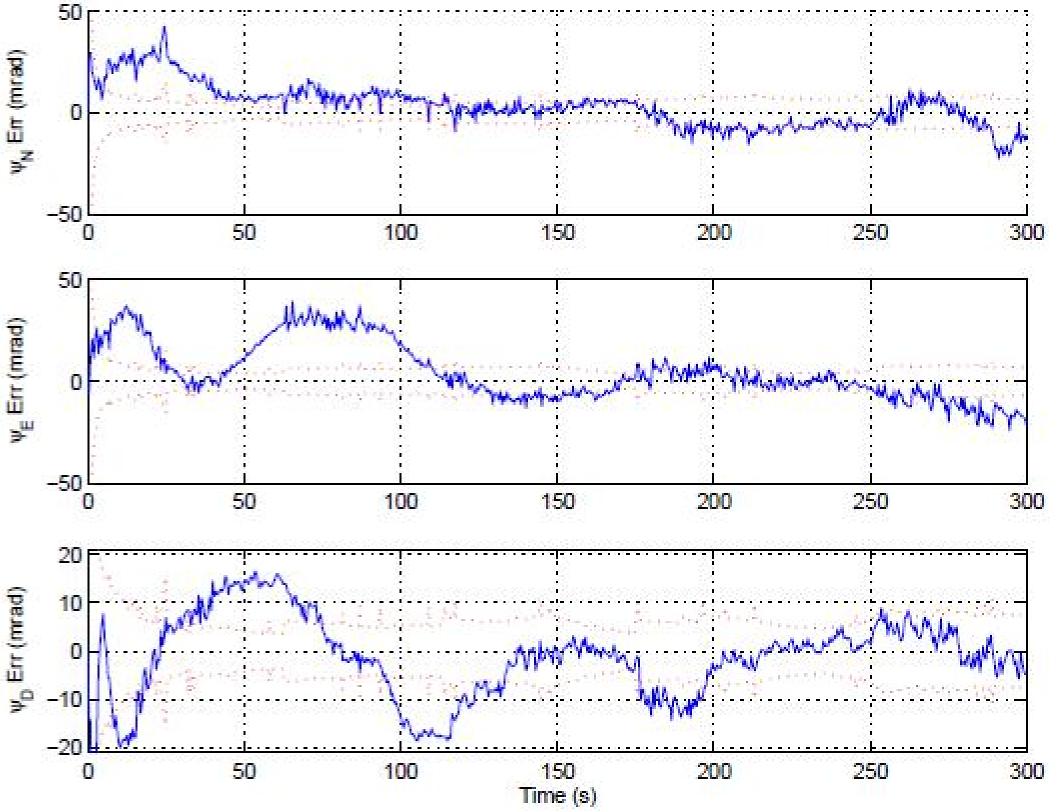


Figure 4.14: The attitude estimates as a result of a previous Monte Carlo simulation, without the algorithm presented in this thesis, are shown here. The dashed lines represent the filter standard deviation and the solid line represents the attitude error while using image aiding (attitude update). Taken from [14].

V. Results

In this chapter, the implications and limitations of the results are discussed. Additionally, potential topics for future research and development are presented.

5.1 Conclusions

All conclusions in this section are pending further adjustments and corrections to the state estimation that reduces or eliminates vulnerabilities to data that forms a solution with singularities. Everything else stated in this section is projected from the current results and should be later confirmed or corrected by any future work. This should not be construed to mean the algorithm is incorrect, so much as it occasionally suffers from numerically unstable computations that cause incorrect results which make it, at this time, undesirable for implementation in a critical navigation system but, in spite of these liabilities, shows great promise for the future.

Based on the RANSAC-based test data presented in Section 4.2, computing the position of an agent such as an aircraft based on matching a priori georegistered map data to the image captured by the agent's camera, it is recommended that the images are transformed to match in either the satellite point of view or into the agent's point of view, but leaving the images in different planes makes comparison more difficult than necessary. From a standpoint of real-time operations, it is much better to transform the image from the agent into the satellite plane, unless the entire flight trajectory including bank angles can be very accurately predicted in advance (in which case, there is little need for a navigation tool such as this). From a navigation perspective, it may be better to transform into the agent's point of view, though the difference in quality between the two perspectives may diminish with further work. It is also possible that such projection is unnecessary given additional benefits from suggested future work.

The most significant contribution is the investigation of the combination of RANSAC and a particle-filter style state estimation algorithm. This demonstrated a very significant improvement over choosing the single best solution with RANSAC, achieving near-truth in some components of attitude. This has the added benefit of harnessing the computation time and results of individual RANSAC iterations that were discarded for not being rated as best.

At this time, incorporating this update into a navigation filter is not recommended, as it will degrade the performance of the filter and worsen the geolocation capability. Pending further work, a significant improvement in accuracy and precision is predicted (see Figures 4.8 and 4.9 for results that experience varying degrees of reliability and Figure 4.10 for some results that are expected to be typical of a stable version). This work has the potential to substitute for GPS when operating in a GPS denied environment.

One of the chief limitations of this work is the necessity of having pre-surveyed image data. In this case, it was generated from satellite observation and georegistration. It is probably possible to generate a computer model of a building from plans and any available images. This technique is definitely applicable indoors if a registered data set can be generated. One of the chief advantages is that it is usable in a GPS restricted or denied environment and it provides an absolute position update otherwise unavailable. Additionally, unlike GPS for a phyically small system, it is able to provide an absolute attitude correction.

The need to enhance the integrity of the data or algorithm used to generate the state estimation has been thoroughly motivated. The next section presents several ways to accomplish this, as well as other ways to improve computation time and navigation reliability.

5.2 Future Work

As with most research, there is room for more work. This section is broken into three parts. The first part is a lower level analysis at potential improvements that could be made to improve the matching portion of the algorithm. The second part builds on that, taking a broader perspective on issues that have overarching navigation concerns rather than simply the matching portion. Finally, this section returns briefly to two of the motivating factors, and offers a way to improve this research with a purely geolocation perspective.

5.2.1 Matching Improvements. Matching presents two significant challenges: getting better matches, and getting matches sooner. That is, increasing the number of positive matches and reducing the run time.

Getting an accurate solution is dependent on successfully devloping a set of positive correct matches. It is not possible to guarantee this, but it is possible to evaluate the quality of the solution developed from any particular set of features and compare it to the traits in both the features and the RANSAC-developed feature transform. While the methods developed in Appendix A helped improve reliability, numerous improvements could be pursued. In the future, these possibilities merit additional work:

- As mentioned in Section 2.8, both SIFT and SURF have additional characteristics that can be used to reliably match, which can be considered an extension of the descriptor. These characteristics were the sign, orientation, and magnitude of the feature. When the sign and magnitude are used, they should be used as a as a constraint. Orientation could be considered as an additional weighting parameter.
- More could probably be done with RANSAC and the associated least squares computation of position and attitude. To reduce or eliminate errors due to poorly conditioned matrices (due to multiple colinear features), an improvement to the current method should be considered. This is potentially the quickest improvement, and would have the greatest impact on the reliability and quality of the solution.
- Another improvement that would reduce the likelihood of a bad position or attitude etimate would be to constrain the accepted estimates to being within

3σ of the propagated inertial solution. Currently, stochastic constraining is only being done on feature location, and not for pose estimation. This would greatly diminish the risk associated with nearly colinear points, which in turn causes an unstable solution. If an image has only a few such nearly colinear feature sets, the interquartile range constraint should eliminate this problem. However, if it has a large number of such sets, such a constraint will be unable to prevent bad solutions from being used in estimation.

- Additional constraints and weighting parameters could be introduced as well, including: consideration of the sign, magnitude, and orientation of the descriptors being matched; the ratio of how often the selected features have been selected compared to the number of times it contributed to a bad solution; and how close the feature is to the horizon (features close to the horizon will likely be poorly described and be poorly located, because the image begins warping severely when the aircraft is too far off nadir).
- Basic properties of the RANSAC-generated transform matrix could be considered, such as the sign and magnitude of the determinant of the RANSAC-generated transform or the conditional number.
- Limit the field of view of the camera when banked. This should clip out the horizon and portions of the terrain in the image that are near the horizon.
- Apply a more complex, but accurate, camera model to capture the effects of distance on the scale of the image (further parts appear smaller).

The current running time of the algorithm presents a significant challenge. The matching operation is, in the current implementation, far and away the most expensive activity. The stochastic constraining is expensive, but it ultimately saves a large number of even more expensive operations, being the comparison of each individual descriptor in the first image to each descriptor in the second image. Improving the run time will allow for a combination of either performing more computations to achieve a better solution, or finishing the computations sooner to achieve a near real-time solution. Some potential ways to improve run time include:

- One path would be to expand the scope of the study on these weights, including analyzing the effect of other factors and re-evaluating all factors as the matches improve, in case one factor emerges as a helpful indicator given other conditions (for example, the spread of features may be a significant indicator more if fewer bad features are being picked from the start).
- The simplest in terms of simulation is to utilize a graphical processor unit (GPU). Though a GPU is currently being utilized when available, it could probably be utilized more. The flip side of this is that it requires a UAV to have such a unit onboard.
- Improving the way the feature database is accessed could provide a sizeable benefit. Currently, it is based on a set of four image tiles, which must have been previously mosaicked together. This imposes a number of unnecessary restrictions that were very convenient for the purpose of providing visual feedback, but ultimately useless to a navigation system. These constraints provide an image-based reference system (the image plane) and transform to combine the four. These components could be abstracted away for a more efficient, more robust system.
- Another way to reduce the running time is to reduce the number of features detected for the image database or the agent. This could be achieved by computing fewer octaves or limiting features based on spatial distribution. The time savigs from using fewer terms in the descriptor could be explored, though this has already been done in [3] and deemed an unlikely path to improve the number of correct positive matches. Though it would improve running time, any loss in reliability should be justified.
- A number of other improvements to the database can be made by partitioning it in accordance with the recommendations on improving reliability of matching.

The recommended constraints encourage a more partitioned data structure. Implementing these should improve the matching reliability and indirectly cut the running time of this portion of the algorithm by more than half if implemented properly. This is illustrated in Figure 5.2.1.

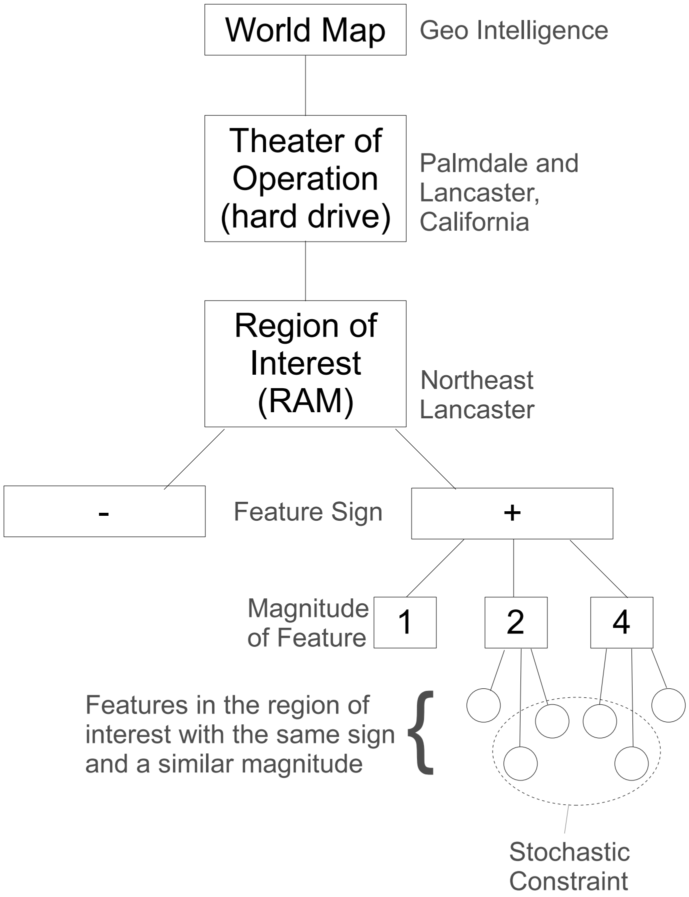

World Map
Geo Intelligence
Theater of Operation (hard drive)
Palmdale and Lancaster, California
Region of Interest (RAM)
Northeast Lancaster
-
Feature Sign
+
Magnitude of Feature
1
2
4
Features in the region of interest with the same sign and a similar magnitude
Stochastic Constraint

Figure 5.1: The general database structure proposed has two substantial benefits: it should substantially speed up the algorithm and make it more precise. It will make it more precise, per [3]. It will speed it up by bisecting the set using the sign and again reducing the set size by using only features within the approximate magnitude. It will then stochastically constrain features by location. This should reduce the number of distance constraint computations as well as descriptor matching computations by a factor of four.

The robustness of the features has room for improvement. Additional work may be done along the lines of the scene model, as utilized in Predictive Based Rendering (see Section 2.9). The solar, geometric, and reflectivity models have the potential to improve positive feature identification. Additionally, it is likely that such models would have a significant impact at lower altitudes or in an urban canyon, where the content in the image cannot be approximated as flat or coplanar. Improvements in feature robustness also increase the reliability of the developed solution.

5.2.2 Navigation Reliability.

The improvements to matching above will provide a more precise solution at closer to real time. Each such improvement makes the navigation more reliable. However, additional changes could be investigated at a higher level than matching that will improve navigation capability.

The simplest change that can be made here is to change the database by expanding it or improving its accuracy. It can include maps at different resolutions (or altitudes) or include a broader area. A more precise map would improve navigation precision by reducing the error in the coordinates of the features stored in the database. Another simple change would be to use a more precise map. In this research, data from Google Maps were utilized. Their maps and coordinate data come from a variety of providers (though it is almost exclusively Digital Globe in the case of this research). Google does not provide warranty on the precision of its coordinates, as it is designed with ground-based navigation in mind, and is meant to get close enough by car, not to guide munitions to a target. Any improvements to the navigation quality will in turn improve the geolocation capability for map generation or targeting.

5.2.3 Geolocation Considerations.

Though geolocation accuracy and precision was a significant motivation for this research, it is only improved indirectly. The work presented here is potentially more useful in that it also improves the navigation solution of the vehicle. However, it is feasible to attempt to short circuit the analysis by locating the object in the camera and simply interpolating based on the matches between the image from the vehicle and the database. A substantial portion of this study would be to investigate the precision of such a technique, as well as what information would be needed to determine its altitude if it is not on the ground.

5.3 Concluding Remarks

Incorporating a georegistration update based on a priori image data has the potential to substantially improve the quality of a navigation estimate and thus the geolocation precision. This is a promising result that could potentially be used to provide precise navigation information even in GPS denied environments.

Appendix A. Tuning RANSAC

This chapter of the appendices explains how the parameters for RANSAC were selected. While this chapter is tangent to the main research thrusts, it was conducted to improve reliability. The way RANSAC is done has the potential to substantially bias the outcome of the main body of research. For that reason, it is worth spending some time studying the effects of the parameters and attempting to give each method the best chance possible.

Because numerous similar words are used that imply a measure, scoring, or other comparison, specific words are now assigned to specific metrics for clarity. Each of these will be described in this chapter at the appropriate place, but Table A is presented early for reference. An extended explanation is also given.

Table A.1: RANSAC Tuning Terminology Summary

Term	Associated Metric or Formula
score	match quality
ratio	match distinctness
SUS value	match selection
weight	any least squares
fitness	RANSAC model quality
threshold	distance constraint

- **score**: The (square of the) distance between two descriptor vectors. In this case, **LOWER** is better.
- **ratio**: The best score divided by the next best score, ratio— $0 < \text{ratio} \leq 1$. **LOWER** is better.
- **SUS value**: The likelihood that two features are a match that will contribute to a good solution. Each element is greater than the last, and the last element is 1 (the first is non-zero). The **GREATER THE DIFFERENCE** between the element and the previous element, the better the match.
- **weight**: Coefficients used to affect the influence one match (measurement) has on a least squares computation. This is used to describe both the RANSAC

model building and model evaluation. **GREATER** coefficients have a greater influence.

- **fitness**: The value assigned to the model generated by RANSAC. In this case, a **LOWER** value is better.
- **threshold** The maximum distance, in pixels, to include a feature in the current RANSAC model.

This chapter is arranged as follows. The first section explains the experiment. The second section explains the change to the random match selector and the rationale for the change. Third, the weighting used for the least squares solution is explained. Fourth, the RANSAC fitness function for model evaluation is presented. Lastly, suggestions for improving this research in the future is presented.

A.1 Analysis of Other Parameters

This section of the appendices discusses the test used to set the RANSAC scoring mechanism. The study was done by running the RANSAC engine, without the fitness test, for 200 attempts over 100 different images for each of the six cases being studied. After each set of matches was made, several pieces of information were recorded:

- score, or distance between descriptor vectors, as recorded by SURF or SIFT
- ratio between the next best match and the best match
- threshold used to determine whether a pixel fits the transform - the RANSAC threshold was randomly generated each sample with a value between 1.5 and 5, which were predicted to be within the optimal operating range.
- error in the position and attitude estimate
- number of matches

Each set was divided again into valid and invalid matches. Valid matches had the following attributes:

- At least three matches were accepted. Fewer can be used in an update, but 3 are needed to fully define the solution.
- A solution was found. The least squares computation is susceptible singularities and gross errors, and when the errors get too large, the computation failed (this failure occurred when trying to generate the overlays shown in Figures 3.4 and 3.5, because the computer would run out of memory by trying to reshape the image incorrectly). This generally would coincide with violating either of the next two criteria.
- No component of the error in the attitude relative to truth was greater than six milliradians.
- The position was more than a kilometer away from where the truth data indicated.

These are rather lax requirements, but the challenge of getting at least three matches with a valid solution is substantial because of the nature of the least squares solution. No SUS values or weights were applied to the solution at this time; all matches were considered equally valid. The formulae developed in this chapter are not guaranteed to be ideal, but are an attempt to properly represent the probability density functions pdfs generated by this experiment. These are used help calculate uncertainty, but do not serve to reject outliers. They were selected by attempting to model the impact of each variable on the graphs shown, either on the chance of success or the chance of failure, whichever was more distinct. This model was then manipulated in each formula to make the best value either the highest or lowest value, as needed.

The rest of this chapter presents plots of the generated probability density functions and any derived formulae for weighting the RANSAC behaviors. All plots are presented as triples. The first plot shows the chance of success against the variable value. The second plot shows the failures, and the last shows the probability that the event occurred at all. The success and failure pdfs are conditional probabilities, both conditioned on the chance that the event occurred at all. Note that this will cause a few curious spikes in the pdfs where the chance of occuring was near, but not at, zero. Tthis causes division by a relatively small fraction, such that the value is magnified artificially; for example, calculating $\frac{1}{1/100}$ will cause a disproportionate result in a section that generally experienced failure.

A.2 Match Selection

Previously, all matches had an equal chance of being selected as the one of the starting four RANSAC seeds. This does not make sense because not all matches are good, and to a good extent, bad matches can be detected in advance.

The final experiment will utilize Stochastic Universal Sampling (SUS), which predicts the quality of each match, and uses that prediction to influence how often it is chosen by using a SUS value. The way this was done is based on the psuedo code, shown below:

```
for $i = 1$ to number of matches do
    $SUSvalue_i = 1/(max(0.25, score_i) \cdot max(0.60, ratio_i))$
end for
$SUSvalue = SUSvalue/sum(SUSvalue)$
for $i = 2$ to number of matches do
    $SUSvalue_i = SUSvalue_i + SUSvalue_{i-1}$
end for
```

The distance between two descriptors is the primary metric used to determine a match. The purpose of this part of the experiment was to see whether the ability to achieve a valid solution was dependent in any capacity on the scores of the various matches incorporated. From visual inspection, it appears that the lower the value of the distance, the better the chance that a good solution would be found. However, the minimum value permitted was 0.25 (the maximum is 4), which is chosen within reasonable bounds of the mathematical justification given in Figure A.1. The minima was imposed to prevent one match from dominating.

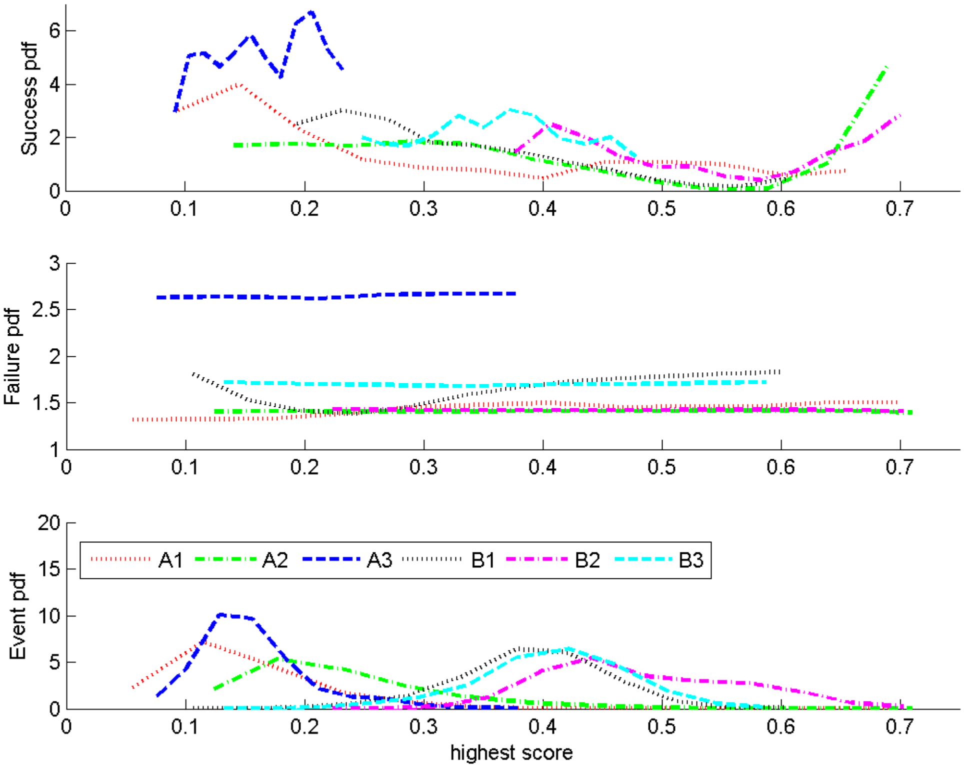


Figure A.1: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the greatest descriptor score. The chance of failure seems to continue even as a match is stronger (strongest on the left), but the chance of success stops sooner (more to the right). Too far to the right, an the chance for success drops off. So, values too far to the left are not necessarily indicative of a better solution, but it would be unwise to penalize a truly good score. To account for this, all values are set to a minimum of 0.25 if they are less than that.

The test of comparing the quality between the best and next best match discussed in [5] is used to remove false matches that occur from noise or unstable effects. The original objective was to determine if thresholding should be done. However, because such a constraint caused SIFT to occasionally and SURF frequently to have no matches, it was instead used in SUS values, weights, and the fitness. Any ratio below 0.6 was set to 0.6 to prevent one match from dominating. As explained in Section 3.4.2, the ratio metric is substantially weakened as a discriminator of good by the small number of matches. However, it is still a good discriminator of bad. The value of 0.6 was selected by inspection from Figure A.2.

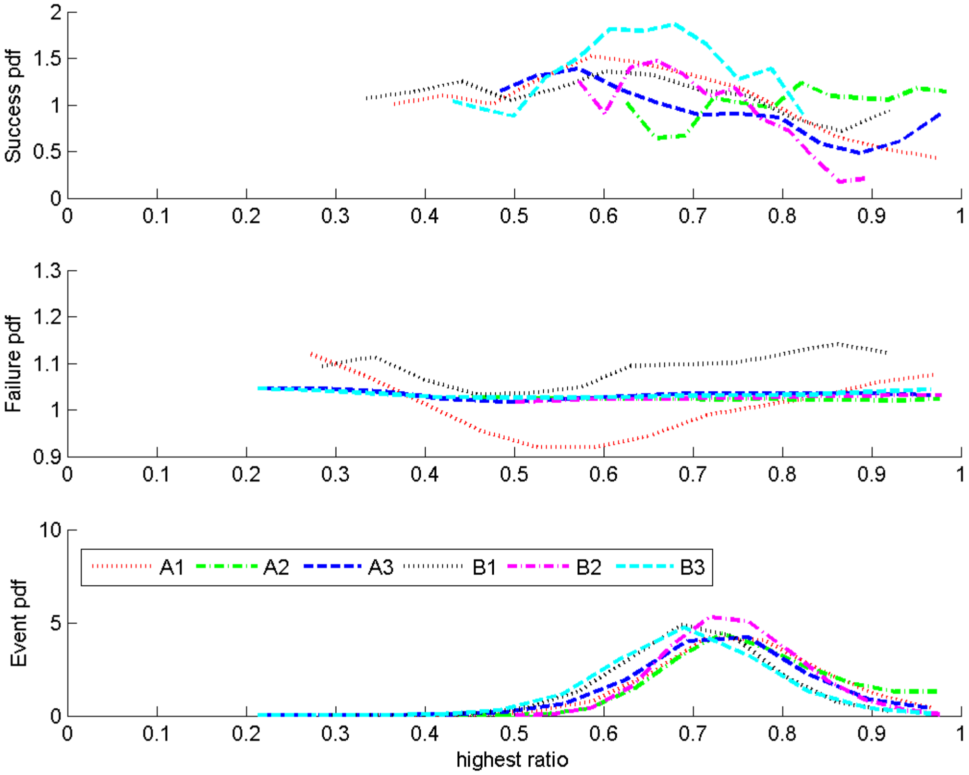


Figure A.2: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the maximum ratio to the next best descriptor. The number of matches likely to be false increased as the quality got too high. The rate of false positives obtained from this indicator should increase drastically as more matches are pruned using stochastic constraints (see Section 3.4.2). The more features that exist to compare against, the stronger this indicator is; by pruning other features, this particular indicator is weakened.

A.3 Least Squares Weighting

When RANSAC determines which matches are within the test model, it generates a transform from the location of features in one image to the other image. Any features that are within the distance threshold set by RANSAC are admitted to the model. However, this can cause bad matches that happen to be in a 'close enough' location to be admitted to the model. The distance between the predicted location and the actual location proved to be a useful, though not absolute, indicator of the likelihood that the match would contribute to a successful or correct solution. To handle this, the least squares solution is weighted to amplify the particles with the highest weight (according to the SUS). The following formula is based on Figure A.3:

$$weight_i = \frac{1}{max(0.25, score_i) \cdot max(0.60, ratio_i)} \left(1 - e^{\frac{-min(1, ProjectiveDistance)}{matches}}\right) \quad \text{(A.1)}$$

Like with the SUS value, the higher the weight, the more influential. The weights were normalized so that the mean was 1. Any model with fewer than three matches was discarded as being an insufficient solution. A minimum model distance of 1 was imposed to prevent a value of zero, which would cause the entire formula to be valued at zero.

RANSAC tends to perform better with a less stringent distance threshold. This is likely a function of the image size and the quality of the transform. Experimentally, it was determined that benefits are realized up to 10 (which was the cap in the experiment), but around 8 was where the benefit seemed to not increase. At some point, the threshold could become so loose that all matches are admitted every time, defeating the purpose of RANSAC. The effect of the distance on the chance of success is demonstrated in Figure A.4.

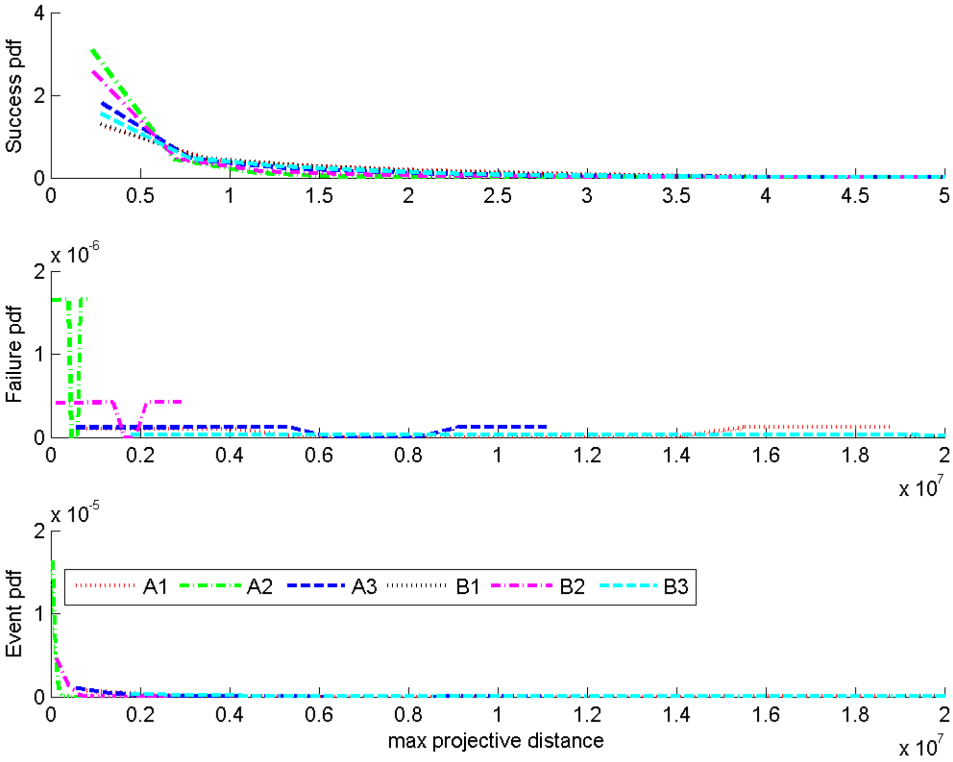


Figure A.3: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the highest projective distance estimated by RANSAC on the overall solution. This indicated that if a feature wasn't virtually colocated with the RANSAC predicted location for it, the chance for success was quite low.

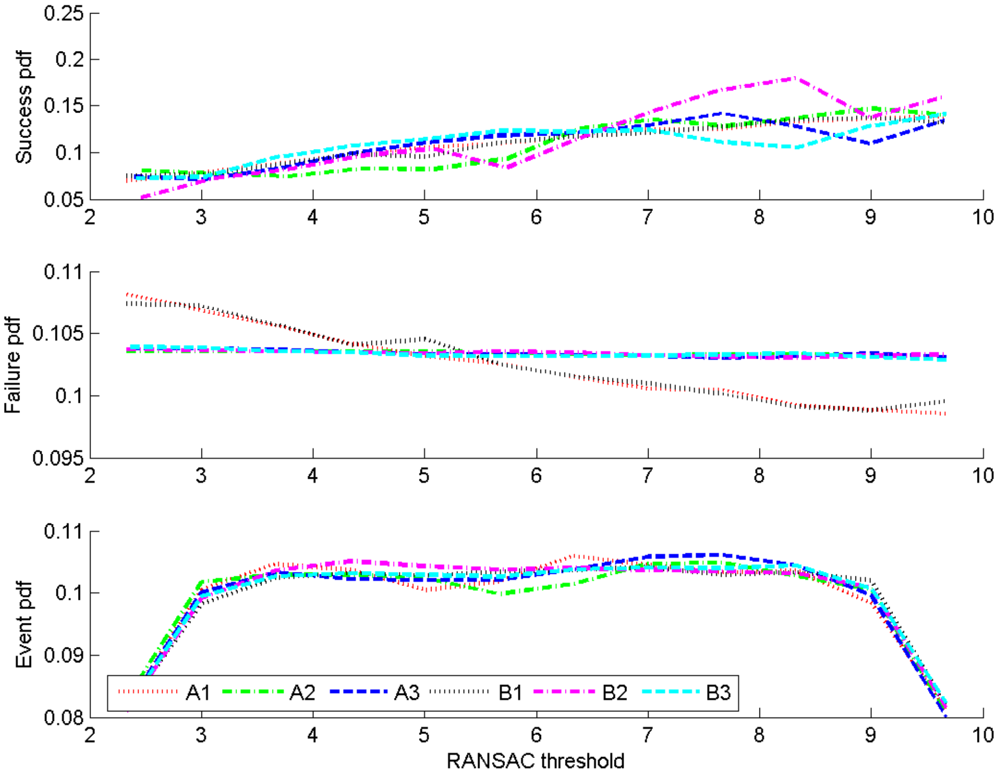


Figure A.4: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the RANSAC distance threshold on the overall solution. By allowing a higher threshold, more values would be accepted from the projective distance computation. This seemed to have an overall positive effect on finding a solution.

A.4 RANSAC Scoring

When individual matches were being weighted for either selection or for least squares, the metrics were based on the worst permissible case. In the case of scoring the overall model, though, what matters most is the mean of these values. It is feasible to re-use the weights again, as to their effect on the mean. This was not done, but would be worth trying in the future.

Four metrics were utilized in scoring an overall model: the mean ratio, the mean score, the number of matches, and the total error in the least squares.

The application from these results in the final run is that at least 3 matches are required and getting more matches is better, to a limit of 7 (i.e. 9 matches are weighted equally with 7). Figure A.5 confirms concerns that more matches does not necessarily guarantee a better solution.

Generally, the lower the mean score was, the more likely that the set would generate a good solution. See Figure A.6.

Likewise, the lower the mean ratio between descriptor distances is, the more likely that the set would generate a good solution. The minimum mean allowed was 0.35, to prevent one from dominating the others. However, due to the earlier constraint on the minimum individual ratio being 0.6, this is guaranteed anyway. Recall that a ratio is only compared against other matches in the stochastically constrained area, meaning that perhaps only one other feature would be present, creating an undeserved low ratio. See Figure A.7.

The residual error from the least squares attempt to generate a model also gives a good indication of the chance of success. To not unfairly bias this towards models having fewer matches, the residual was divided by the total number of matches. See Figure A.8.

The formula used to weight a model is as follows, with the same minimum score of 0.25 and ratio of 0.60 for each match used:

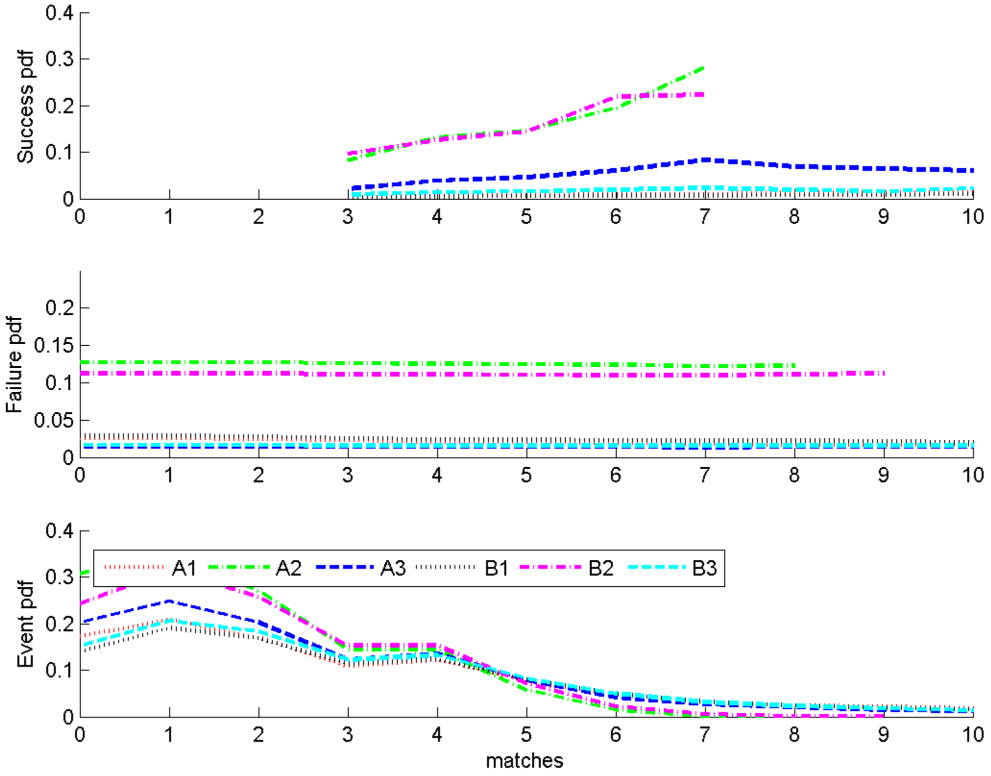


Figure A.5: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the impact of the number of matches used in a solution. More matches generally increased the chance of success. A minimum of three matches were needed for success.

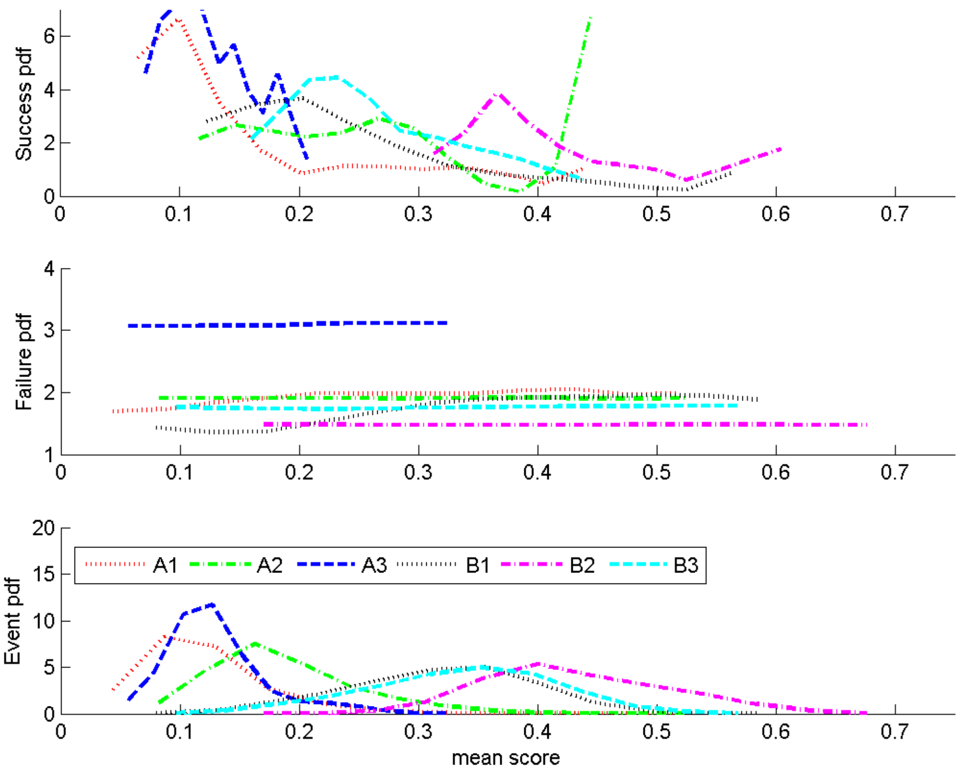


Figure A.6: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the mean score of all matches. Generally, the lower the mean score of all the matches in a solution set are, the more likely it was to be successful.

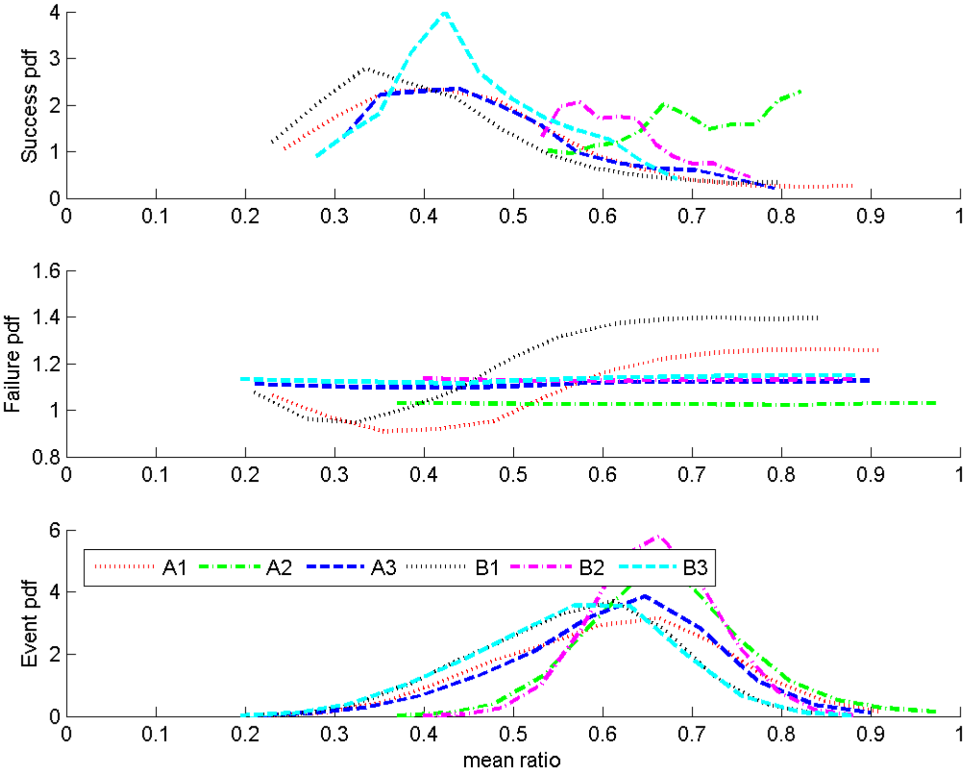


Figure A.7: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the effect of the mean ratio. As the ratio decreases, the chance of success increases, but only to a point; after that point, it decreases some. While it would be unwise to penalize a good result, the quality of the ratio indicator is reduced by the smaller set of features to compare against.

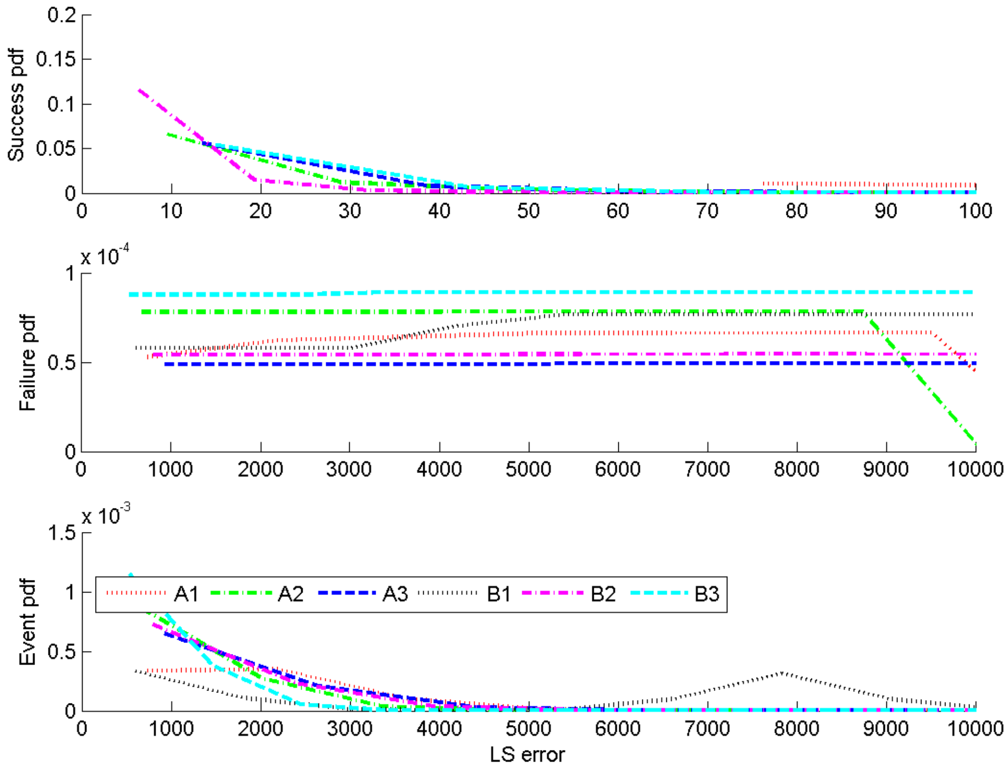


Figure A.8: This figure shows, from top to bottom, the probability that a solution was good; the probability that a solution was bad; and the distribution of the data set regarding the impact of least squares residual error. A lower error generally indicated an increased likelihood of a successful match.

$$fitness = mean(score) \cdot mean(ratio)) \cdot \tfrac{1}{ln(matches)} \cdot \left(1 - e^{\frac{-residual_{LS}}{matches}}\right)$$

In this case, the lowest fitness was best.

Bibliography

1. “Digital Globe”. Website, 2009. Available at www.digitalglobe.com.

2. “National Geospatial-Intelligence Agency”. Website, 2009. Available at www1.nga.mil.

3. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded Up Robust Features”. Jauary.

4. Fishcler, Martin A. and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. *Communications of the ACM*, 24(6):381–395, June 1981.

5. Lowe, David G. “Distinctive Image Features from Scale-Invariant Keypoints”. *International Journal of Computer Vision*, January 2004.

6. Maybeck, Peter S. *Stochastic Models, Estimation, and Control Volume 1.* Navtech Book and Software Store, Arlington, VA, 1994.

7. Michael J. Veth (Major, USAF). *Fusing of Imaging and Inertial Sensors for Navigation.* Ph.D. thesis, Air Force Institute of Technology, 2006.

8. Mikolajczyk, Krystian and Cordelia Schmid. “A Performance Evaluation of Local Descriptors”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.

9. Moigne, Jacqueline Le, James Carr, Donald Chu, and Jaime Esper. “Advanced Geosynchronous Studies Imager (AGSI): Image Navigation and Registration (INR) System”. *Conference on Earth Observing Systems IV*, volume 3750, 23–34. The International Society for Optical Engineering (SPIE), 1999.

10. NIMA. *Department of Defense World Geodetic System 1984.* Technical Report NIMA TR8350.2, National Imagery and Mapping Agency NIMA, 2000.

11. Purman, Benjamin, James Spencer, and Jennifer M. Conk (1st Lt USAF). “Prediction-Based Registration: An Automated Multi-INT Registration Algorithm”. *Algorithms for Synthetic Aperture Radar Imagery XI*, volume 5427, 249–258. The International Society for Optical Engineering (SPIE), 2004.

12. Richard Radke, Tomio Echigo, Peter Ramadge. “Efficiently Estimating Projective Transforms”. June 2001.

13. Titterton, David H. and John L. Weston. *Strapdown Inertial Navigation Technology.* Institute of Electrical Engineers and American Institute of Aeronautics and Astronautics, Bodmin UK, second edition, 2004.

14. Veth, Michael, Robert C. Anderson, Fred Webber, and Mike Nielsen. *Tightly-Coupled INS, GPS, and Imaging Sensors for Precision Geolocation.* DTIC

ADA478300, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45324, JAN 2008.

15. Zogg, Jean-Marie. *GPS Essentials of Satellite Navigation Compendium.* GPS-X-02007-D. u-blox AG, Bodmin UK, 2008.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 10 Sep 09	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Oct 2007 — Sept 2009

4. TITLE AND SUBTITLE

Precision Navigation Using Pre-Georegistered Map Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Webber, Frederick C.

5d. PROJECT NUMBER

No Funds

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/09-54

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory / RYRN (Dr. Jacob L. Campbell)
Bldg 620, ROOM 3AJ39
2241 Avionics Circle
WPAFB, OH 45433-7333
jacob.campbell@wpafb.af.mil (937) 255-6127, ext 4154

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYRN

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Navigation performance in small unmanned aerial vehicles (UAVs) is adversely affected by limitations in current sensor technology for small, lightweight sensors. Because most UAVs are equipped with cameras for mission-related purposes, it is advantageous to utilize the camera to improve the navigation solution. This research improves navigation by matching camera images to a priori georegistered image data and combining this update with existing image-aided navigation technology. The georegistration matching is done by projecting the images into the same plane, extracting features using the techniques Scale Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF). The features are matched using the Random Scale and Consensus (RANSAC) algorithm, which generates a model to transform feature locations from one image to another. In addition to matching the image taken by the UAV to the stored images, the effect of matching the images after transforming one to the perspective of the other is investigated. One of the chief advantages of this method is the ability to provide both an absolute position and attitude update.

15. SUBJECT TERMS

navigation, artificial intelligence, image processing, feature extraction, georegistration, geolocation, image matching, perspective transformation, pose estimation, RANSAC, kalman filtering

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			LtCol Michael J. Veth, PhD
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	117	(937) 255–3636, ext 4541; Michael.Veth afit.edu